Fault Equivalence, Dominance & Collapsing

- Definition: If T_a is the set of *ALL* TVs which Detect Fault *a*, and T_b is the set of *ALL* TVs which Detect some other Fault *b*; the Two Faults *a*, and *b* are said to be ***Equivalent*** IFF $\mathbf{T_a = T_b}$.

- In Other Words, *Two Faults* are ***Equivalent*** IFF *any* test detecting *one*, *also* detects *the Other* and vice versa

Fault Equivalence

- Lemma: If faults *a* and b are equivalent then the corresponding faulty functions F_a and F_b are identical.
- Let $t1 \in \mathbf{T_a} \rightarrow t1 \in \mathbf{T_b}$ (*Since* $\mathbf{T_a = T_b}$)
- $F(t1) \oplus F_a(t1) = 1$ && $F(t1) \oplus F_b(t1) = 1$
- The above Equations are Valid $\forall\, t1 \in \mathbf{T_a}$ **(=** $\mathbf{T_b}$**), Thus**
- $[F(t1) \oplus F_a(t1)] \oplus [F(t1) \oplus F_b(t1)] = 0 \rightarrow F_a(t1) \oplus F_b(t1) = 0$
- Thus $F_a = F_b$
- **Definition**: *Two Faults* are ***Equivalent*** IFF *They Result in the Same Faulty Function*

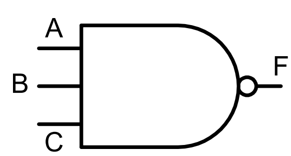
Fault Equivalence

- Fault Equivalence is an Equivalence Relationship, i.e. it is
  - Reflexive → $T_a = T_a$
  - Symmetric ($T_a = T_b$ → Reflexive → $T_b = T_a$)
  - Transitiveive : Thus

    ($T_a = T_b$ && $T_b = T_c$ → Reflexive → $T_a = T_c$)

- ***Thus, Equivalent Faults are Grouped into Equivalence Classes***
- ***Only One Fault of Each Class Need Be Included In the Fault List***

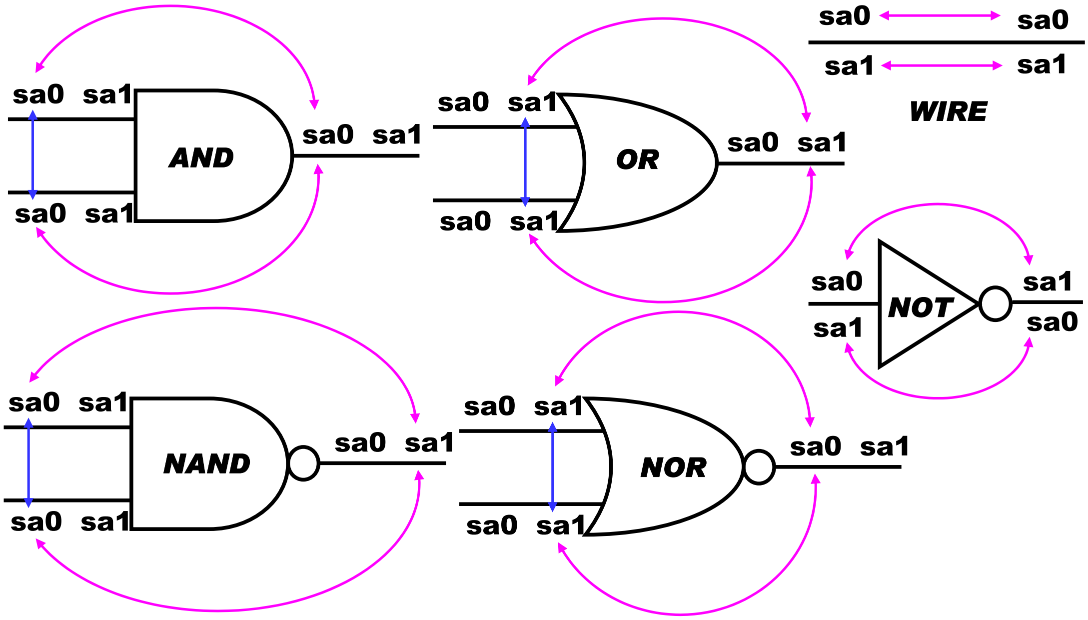
Fault Equivalence (Example)



3-Input AND Gate

- ***Faults*** **(** ***A/0 , B/0, C/0, F/1*** **) are All Detectable by the single TV (** ***t = 111/0*** **) → All 4 Faults Are Equivalent**

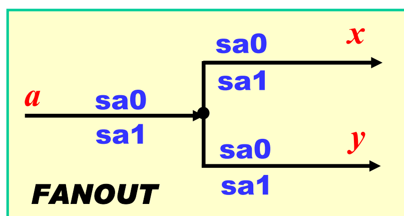
Equivalence Rules



Fault Equivalence Indicated By Blue or Violet lines with Two-Way Arrows

Equivalence Rules

For, Multi-Output Functions, All Faulty Outputs Must Be Identical to Establish Equivalence



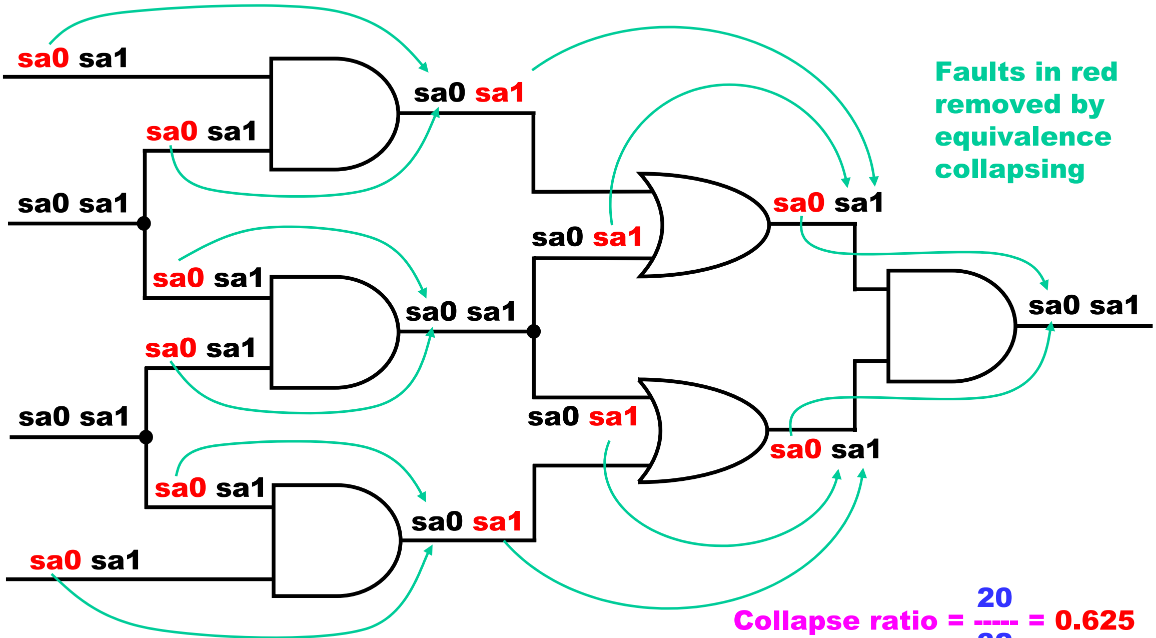
Multiple Fault

Fault	Test (a → x, y)	Faulty Output Functions
a/0	1→ 1/0 , 1/0	(*x* = *0*, *y*=*0*)
x/0	1→ 1/0 , 1	(*x* = *0*, *y*=***a***)
y/0	1→ 1, 0/1	(*x* = ***a***, *y*=*0*)
x/0 && y/0	1→ 1/0 , 1/0	(*x* = *0*, *y*=*0*)

Single Faults at the STEM and Fanout Nodes

Are Not Equivalent

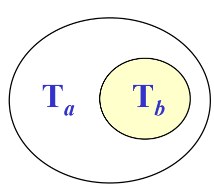
Equivalence Example



Collapse ratio = $\frac{20}{32}$ = 0.625

Fault Dominance

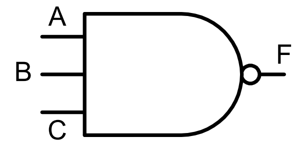
- Definition: If T_a is the set of *ALL* TVs which Detect Fault *a*, and T_b is the set of *ALL* TVs which Detect some other Fault *b*; Fault *a* is Said to Dominate Fault *b* IFF $\mathbf{T}_b \subseteq \mathbf{T}_a$.



- In other words, IF all TVs of some fault *b* also detect another fault *a*, then *a* is said to dominate *b*.
- If two faults dominate each other then they are equivalent.

Fault Dominance (Example)

3-Input AND Gate



Fault	Detecting TVs
A/1	011/1
B/1	101/1
C/1	110/1
F/0	{011/1,101/1,110/1, 010/1, 001/1,100/1, 000/1}

Obviously F/0 Dominates A/1, B/1 and C/1

Note Dominant Faults Need Not Be Included in The Fault List

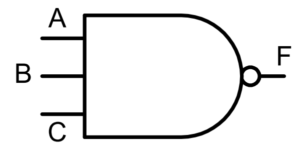
Fault Collapsing

- Only One Fault From Each Class of Equivalent Faults Need Be Included In The Fault list (*Equivalence Fault Collapsing*)
- Only Dominated Faults Need Be Included In the Fault list (*Dominance Fault Collapsing*)
- For an *n*-input gate, at most *n+1* faults need to be considered out of the Possible 2(n+1) Faults (exactly *n+1* faults for AND, OR, NAND, NOR, and Inverter Gates).

Fault Collapsing (Example)

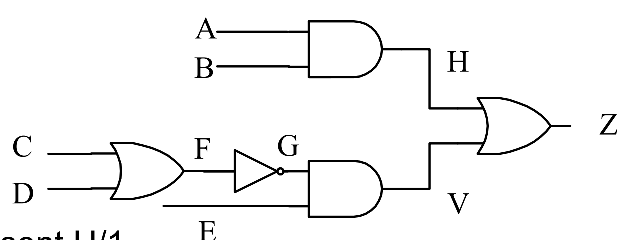
<u>3-Input NAND Gate</u>

TV ABC F	Detected Faults
1 1 1 0	A/0≡B/0 ≡ C/0 ≡ F/1
0 1 1 1	A/1 ⊂ F/0
1 0 1 1	B/1 ⊂ F/0
1 1 0 1	C/1 ⊂ F/0



- For NAND Gate with *n* I/Ps, Only (*n+1*) Faults Need Be Considered
- When dominance fault collapsing is used, it is sufficient to consider only the input faults {A/0, A/1, B/1, C/1}

Fault Collapsing Example



1 {*A*/0, *B*/0, *H*/0}
2 {*C*/1, *D*/1, *F*/1, *G*/0}
3 {*E*/0, *G*/0, *V*/0}
4 {*H*/1, *V*/1, *Z*/1}
5 {*F*/0, *G*/1}
6 *A*/1 ⊂ *H*/1, thus *A*/1 can represent H/1 and all its equivalent faults in class 4
7 *C*/0 ⊂ *F*/0, thus *C*/0 can represent F/0 and all its equivalent faults in class 5
8 V/0 ⊂ *Z*/0, but V/0 belongs to equivalence class 3, which has been merged into class 2. Any fault from this class is dominated by Z/0.
9 *B*/1 ⊂ *H*/1
10 *D*/0 ⊂ *F*/0
11 *E*/1 ⊂ *V*/1

- **{*A*/0, *A*/1, *B*/1, *C*/0, *C*/1, *D*/0, *E*/1}**

In a **Fanout-Free** circuit (Tree-like with no fanout nodes), PI faults form a dominance collapsed fault set.

Fanout-Free Circuits

- This is a special class of circuits that are easy to test

Definition 1: Every line x has a maximum fanout of one gate.

Definition 2: There is just one path from every line x to the primary output

- Every logic circuit can be decomposed into a set of fanout-free subcircuits separated by fanout points

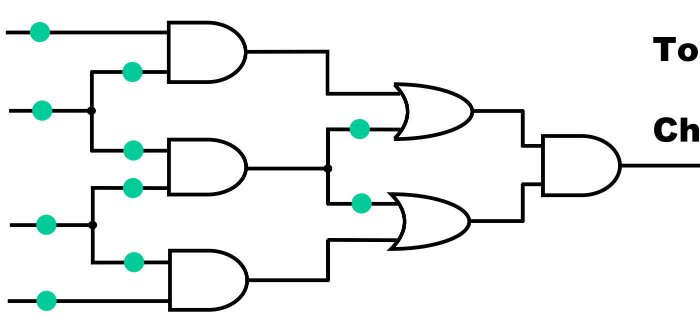
Definition 3: A set of tests $T = \{t_1, t_2, \ldots, t_k\}$ is ***complete*** if it detects (covers) all detectable SSL faults in the circuit

Theorem

- In an fanout-free circuit, every test set that detects all SSL faults on primary inputs is ***complete*** .

Checkpoints

- Primary inputs and fanout branches of a combinational circuit are called *checkpoints*.
- **Checkpoint theorem**: A test set that detects all single (multiple) stuck-at faults on all checkpoints of a combinational circuit, also detects all single (multiple) stuck-at faults in that circuit.



Total fault sites = 16

Checkpoints (●) = 10

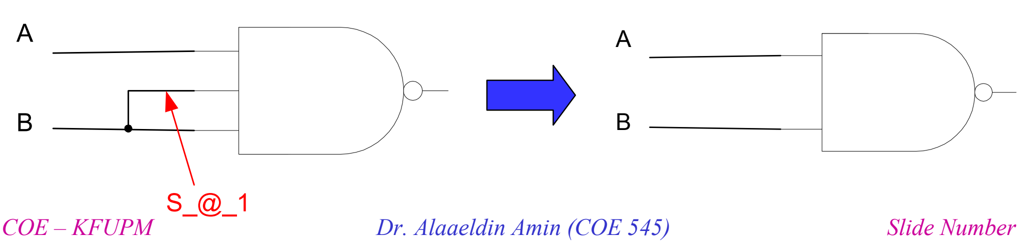
Classes of Stuck-at Faults

- Following classes of single stuck-at faults are identified by fault simulators:
  - *Potentially-detectable fault* -- Test produces an unknown (X) state at *primary output* (PO); detection is probabilistic, usually with 50% probability.
  - *Initialization fault* -- Fault prevents initialization of the faulty circuit; can be detected as a potentially-detectable fault.
  - *Hyperactive fault* -- Fault induces much internal signal activity without reaching PO.
  - *Undetectable*(*Redundant) fault* -- No test exists for the fault.
  - *Untestable fault* -- Test generator is unable to find a test.
  - Some faults are detected by lots of tests—"easy" faults
  - Some faults are detected by few tests—"hard" faults
  - Some input patterns detect no faults
  - The number of tests needed for a *complete test set* is often a small fraction of the total number of input patterns available
  - Finding a minimal test set is hard and usually impractical

Combinational Circuit Testing

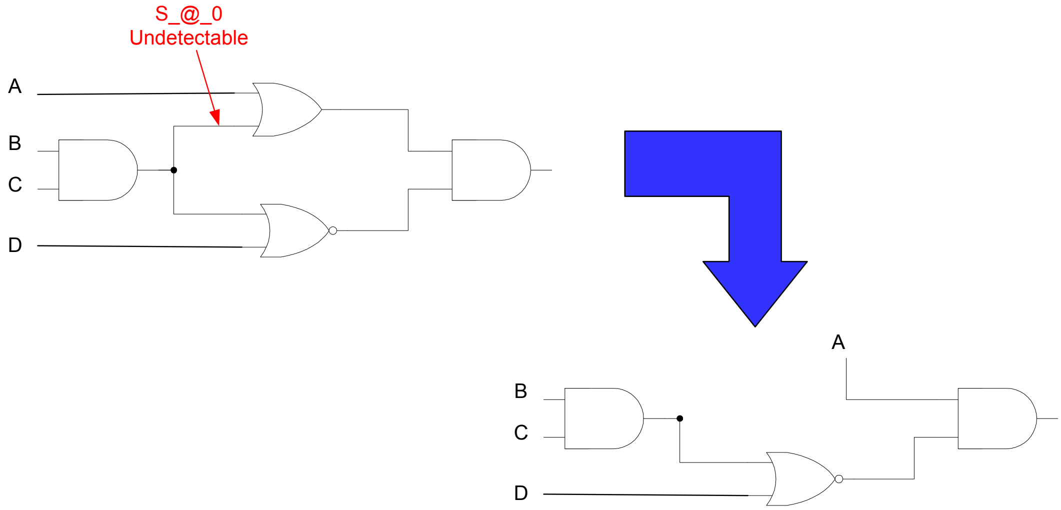
Basic Definitions

- Let fault *f* change output Z(X) of a circuit *C* to $Z_f(X)$.
- A TV *t* detects *f* if $Z(t) \neq Z_f(t) \rightarrow Z(t) \oplus Z_f(t) = 1$
- Fault *f* is Undetectable or Redundant if $Z(t) = Z_f(t)\ \forall t$.
- If the fault *line x s-a-d* is undetectable, then *x and circuits feeding x* can be removed from the circuit.

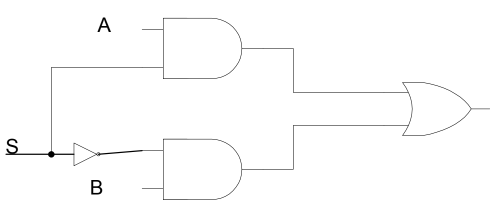


Combinational Circuit Testing

Another Example

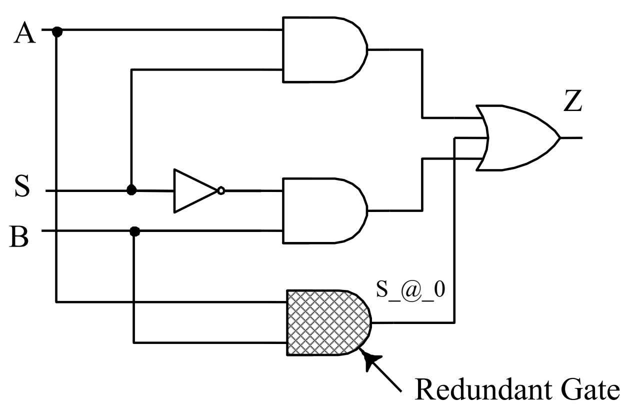


Redundancy



Hazardous Circuit

(a)



Hazard-Free Circuit

(b)

Combinational Circuit Testing

Basic Definitions (cont'd)

- A set of tests $T = \{t_1, t_2, \ldots, t_k\}$ is *complete* if it detects (covers) all detectable SSL faults in the circuit
- We can represent any test set $\{t_1, t_2, \ldots, t_k\}$ by the Boolean function whose minterms are $\{t_1, t_2, \ldots, t_k\}$.
- Example, the function $Z(a,b,c,d) = a'bd' + abcd$ denotes the 3-member test set {0100, 0110, 1111}.
- The set of all tests for fault f is expressed by the Boolean function $(Z(x) \oplus Z_f(x))$

Boolean Difference

- **Shannon's Expansion Theorem:**
  $F(X_1, X_2, \ldots, X_n) = X_2 \bullet F(X_1, 1, \ldots, X_n) + \overline{X_2} \bullet F(X_1, 0, \ldots, X_n)$
- **Boolean Difference (partial derivative):**

$$\frac{\partial F_j}{\partial g} = F_j(1, X_1, X_2, \ldots, X_n) \oplus F_j(0, X_1, \ldots, X_n)$$

- **Fault Detection Requirements for g stuck-at 0:**

$$G(X_1, X_2, \ldots, X_n) = 1$$

$$\frac{\partial F_j}{\partial g} = F_j(1, X_1, X_2, \ldots, X_n) \oplus F_j(0, X_1, \ldots, X_n) = 1$$