Fault Detection of VLSI/SoC/IC Testing

Dr. Katherine Shu-Min Li
EDA&T Lab.
Dept. Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, TAIWAN

Thanks to Prof. Sying-Jyan Wang, sjwang@cs.nchu.edu.tw

Review:
How To Do Test: **FIVE** Approaches

- **Fault Modeling**
  - Identify **target faults**
  - Limit the **scope of test generation**
  - Make analysis possible
- **Test Generation**
  - Automatic or Manual
- **Fault Simulation**
  - Assess completeness of tests (test patterns), fault/test coverage
- **Testability Analysis**
  - Analyze a circuit for potential problems on test generation
- **Design For Testability**
  - Input for **Controllability**; Output for **Observability**
  - Design a circuit for guaranteed test generation
  - Introduce both area overhead and performance degradation

Review: Fault Modeling

- **Fault Models**
  - **Stuck-At Faults**
  - **Bridging Faults**
  - **Transistor Stuck-On/Open Faults**
  - **IDDQ Faults**
  - **Functional Faults**
  - **Memory Faults**
  - **PLA Faults**
  - **Delay Faults & State Transition Faults**

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

Why **Logical** Fault Modeling

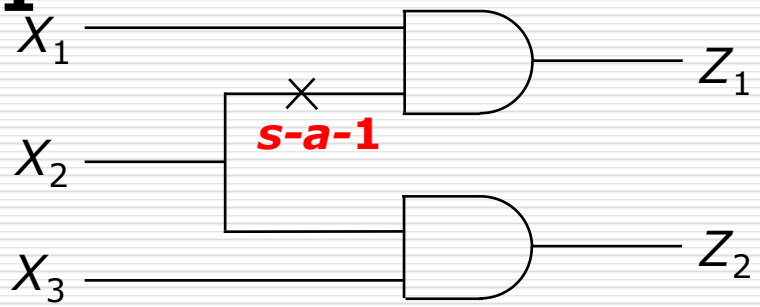
- Fault analysis on logic rather than physical defect problem
  - **Complexity is reduced**
- **Technology independent**
  - Same fault model is applicable to many technologies
  - Testing and diagnosis methods remain valid despite changes in technology
- **Tests/Test Inputs** derived may be used for physical faults
  - Physical faults/defects effect on circuit behavior is not completely understood or too complex to be analyzed
- **Stuck-at fault**: The most popular logical fault model

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

Fault Detection

- A **test(vector) *t*** detects a fault ***f*** if and only if
  - *t* detects *f* $\Leftrightarrow$ $z_f(t) \neq z(t)$
- Example
  - *t*=(100) detects *f* because $z_1(100)=0$ and $z_{1f}(100)=1$



$z_1 = x_1 x_2$ $\quad Z_2 = X_2 X_3$

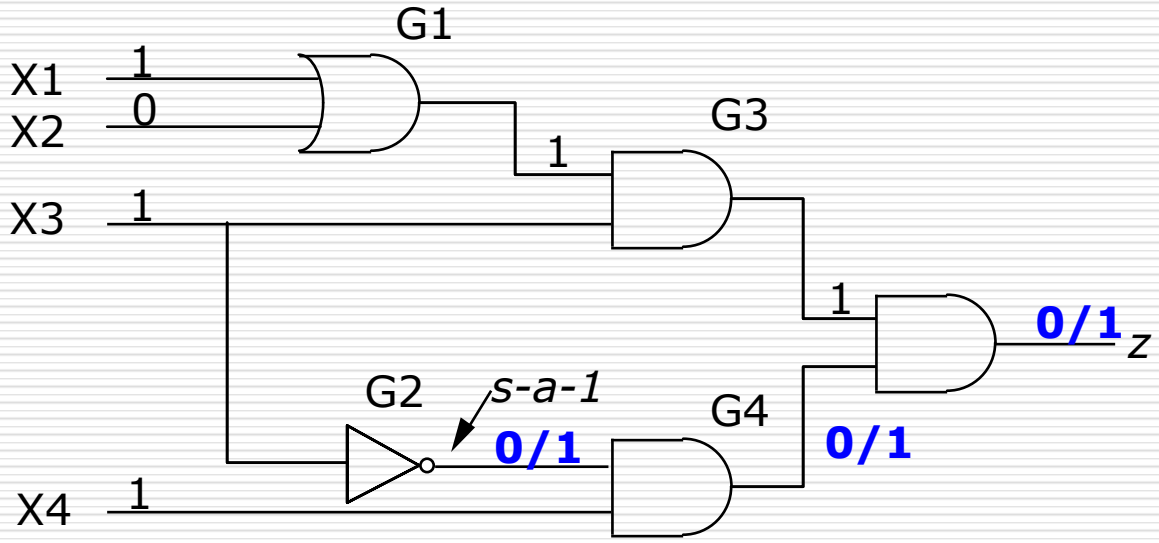
$z_{1f} = x_1$ $\quad Z_{2f} = x_2 X_3$

Summary: X2=X2_stem + X2_up_fanout(sa1, s-a-1)+X2_down_fanout

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

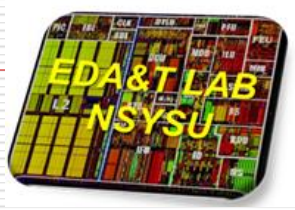
Sensitization (1/2)



$Z(1011) = 0$

$Z_f(1011) = 1$

- 1011 detects the fault f (**G2 stuck-at 1**)
- v/v_f
  - v = signal **v**alue in the **fault free** circuit
  - v_f = signal **v**alue in the **faulty** circuit



Sensitization (2/2)

- A test/test input *t* detects a fault *f*
  - **Step 1. FA: Activate *f*** (or **generate a fault effect**)
    - by creating different v and v_f values at the site of the target fault => $\boldsymbol{v/v_f}$
    - =>Test Input=>Controllability
  - **Step 2. FP: Propagate** the error to a **primary output *w***
    - by making all the lines along **at least one path between the fault site and *w***
    - *w* have different v and v_f values
    - =>Observability
- A line whose value in the test/test input changes in the presence of the fault *f* is said to be **sensitized to the fault *f* by the test/test input**
- A path composed of sensitized lines is called a **sensitized path**

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

Detectability

- A fault *f* is said to be **detectable** if there exists **a test/test input *t* that detects *f*** ; otherwise, *f* is an undetectable fault
- For an **undetectable** fault *f*
  - **No test** can simultaneously **activate *f*** and **create a sensitized path** to a **primary output**
  - **Step 1 Fault Activation and Step 2 Fault Propagation (sensitization) need to be satisfied at the same time**

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

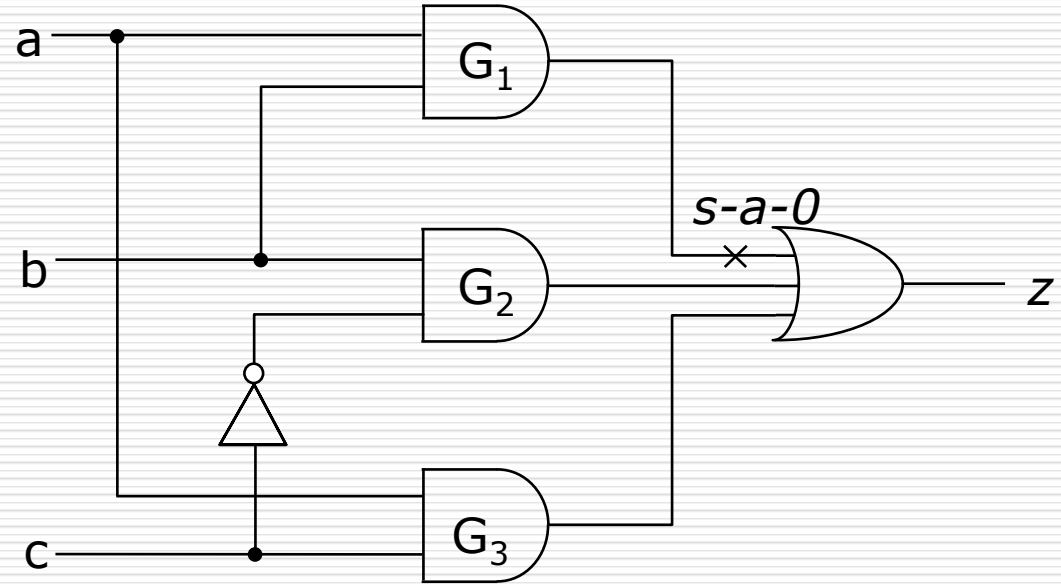
Undetectable Faults (1/3)

- A fault *f* is said to be detectable if there exists a test *t* that detects *f*; otherwise, *f* is an undetectable fault
- For an **undetectable** fault *f*

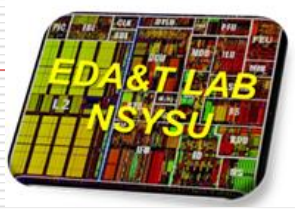
$$z_f(x) = z(x) \quad \textbf{for all } x$$

  - No test can simultaneously activate *f* and create a sensitized path to a primary output

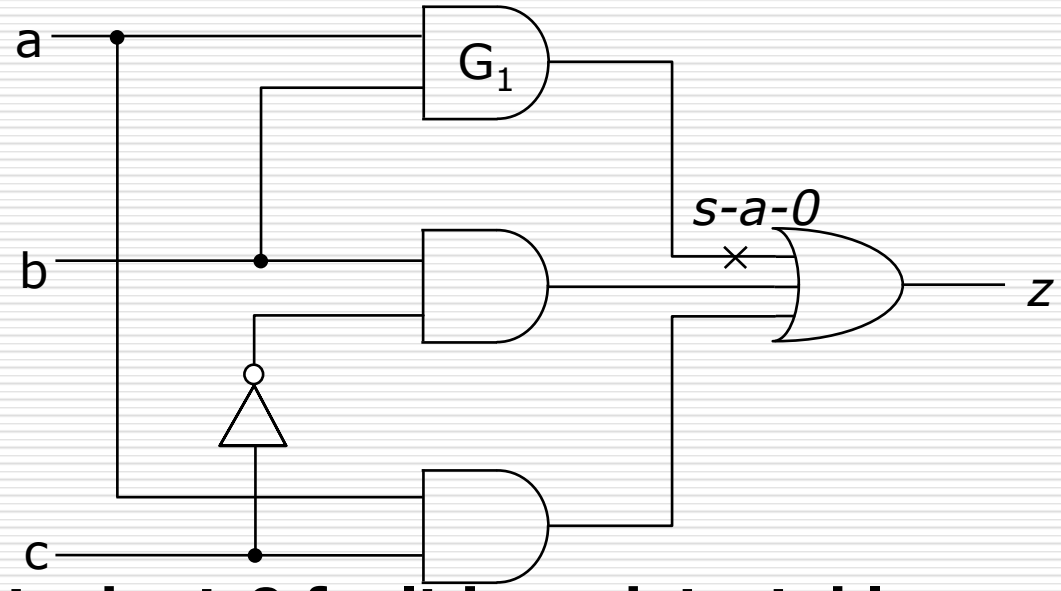
Undetectable Faults (2/3)



- G_1 output stuck-at-0=>**1/0**
- Zf=OR(G1,G2,G3)=1, Z=OR(G1,G2,G3)
- **G2=0 and G3=0 and G1=ab=1**
- **G1=ab=1=>a=1 and b=1**
- **G2=b*inv(c)=0=>c=1** and **G3=ac=0=>c=0**
- Thus, **conflict at c=>No test vector available**



Undetectable Faults (3/3)



- **G_1 output stuck-at-0 fault is undetectable**
  - Undetectable faults **do not change the *function* of the circuit**
    - **According to *Truth Table (K-map, Karnaugh Map)***
  - The **related circuit can be deleted** to simplify the circuit
  - **s-a-0 just be replaced by Line 0; s-a-1 by Line 1**
- **Undetectable faults = Redundant faults**

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
  - **Detectable vs Undetectable**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

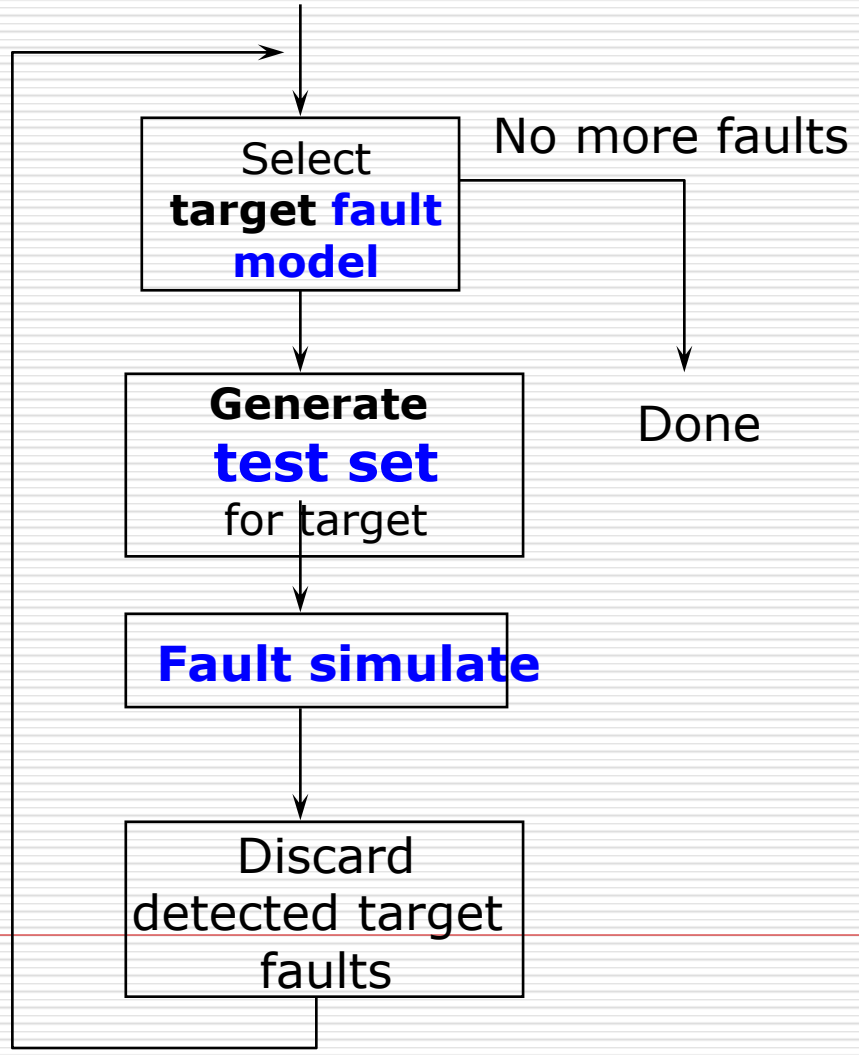
Test Set

- **Complete detection test set**: A set of tests/test inputs that detect any detectable faults in a class of faults
- **Test Quality**: The **quality of a test set** is measured by **test/fault coverage**
- **Fault coverage**: Fraction of faults that are detected by a test set
- The fault coverage can be determined by **fault simulation**
  - >95% is typically required for **single stuck-at fault** model
  - >99.9% in IBM
  - Four 9 principle: 99.99%

Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
  - **Detectable vs Undetectable**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

Typical **Test Generation Flow**



Fault Detection

- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
  - **Detectable vs Undetectable**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

Fault Equivalence (1/3)

- A test t **distinguishes** between faults α and β if $z_\alpha(t) \oplus z_\beta(t) = 1$
- Two **faults**, α & β are said to be **equivalent** in a circuit , iff the **function under** α is equal to the **function under** β for any input combination (sequence) of the circuit.
  - $z_\alpha(t) = z_\beta(t)$ **for all** t
  - **No test** can distinguish between α and β
  - Any test which detects one of them detects all of them

Fault Equivalence (2/3)

- **AND** gate
  - All *s-a-**0*** faults are equivalent
- **OR** gate
  - All *s-a-1* faults are equivalent
- **NAND** gate
  - All the **input** *s-a-**0*** faults and the **output** *s-a-1* faults are equivalent
- **NOR** gate
  - All **input** *s-a-1* faults and the **output** *s-a-0* faults are equivalent
- **Inverter**
  - **Input *s-a-1*** and **output *s-a-0*** are equivalent
  - **Input *s-a-0*** and **output *s-a-1*** are equivalent

- Key concept: all **CV (Controlling Value)** are equivalent

Fault Equivalence (3/3)

GATE Type	CV (Controlling Value)	NCV (Non-Controlling Value)
AND	0	1
NAND	0	Only output inv 1
OR	1	0
NOR	1	Only output inv 0
INV	0 vs 1; 1 vs 0	
XOR XNOR	0 vs 1;1 vs 0 0 vs 0;1 vs 1	XOR=1 for different inputs XNOR=1 for Same inputs

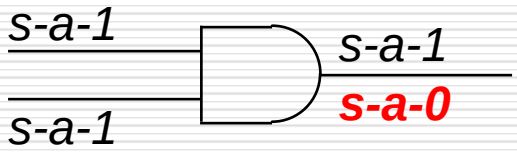
Fault Detection

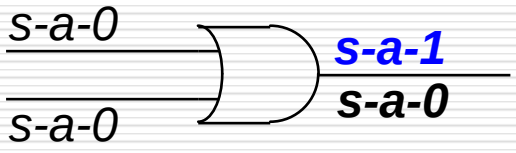
- **Why Logical Fault Modeling**
- **Fault Detection**
- **Sensitization**
- **Detectability**
  - **Detectable vs Undetectable**
- **Undetectable Faults**
- **Test Set**
- **Test Generation Flow**
- **Fault Equivalence**
  - **Equivalence Fault Collapsing**

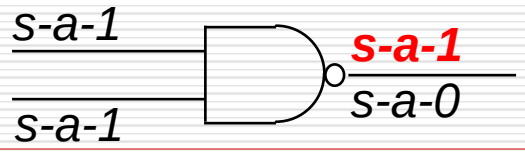
Equivalence Fault Collapsing (1/3)

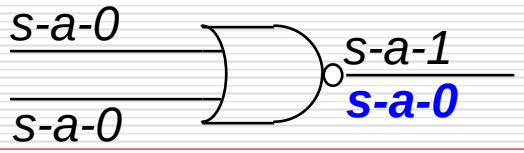
(N)AND vs (N)OR

- ***n inputs, 1 output under SA0/SA1 =>2(n+1)=2+2***
- ***By Controlling Value: only ONE from n-inputs & 1 output is enough=>delete #n faults***
  - ***n+2*** instead of ***2n+2*** faults need to be considered for ***n*-input gates**









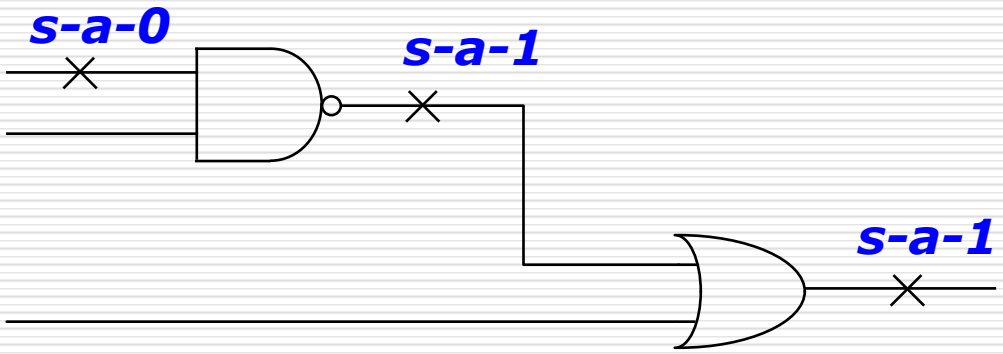
Equivalence Fault Collapsing (2/3)

Wire

- Equivalence in a wire [between two endpoints]
  - Fault equivalence
    - **A sa0 ↔ B sa0**
    - **A sa1 ↔ B sa1**
  - No need to consider B sa0 and B sa1
  - Summary: Both ends of a wire are fault equivalence because same voltage value

Equivalence Fault Collapsing (3/3)

- Two/Multiple **equivalent faults are detected by exactly the same tests**
  - The following three faults are equivalent
  - That is, **same FEs along propagation path**



Thank You for Your Attention & Participation!