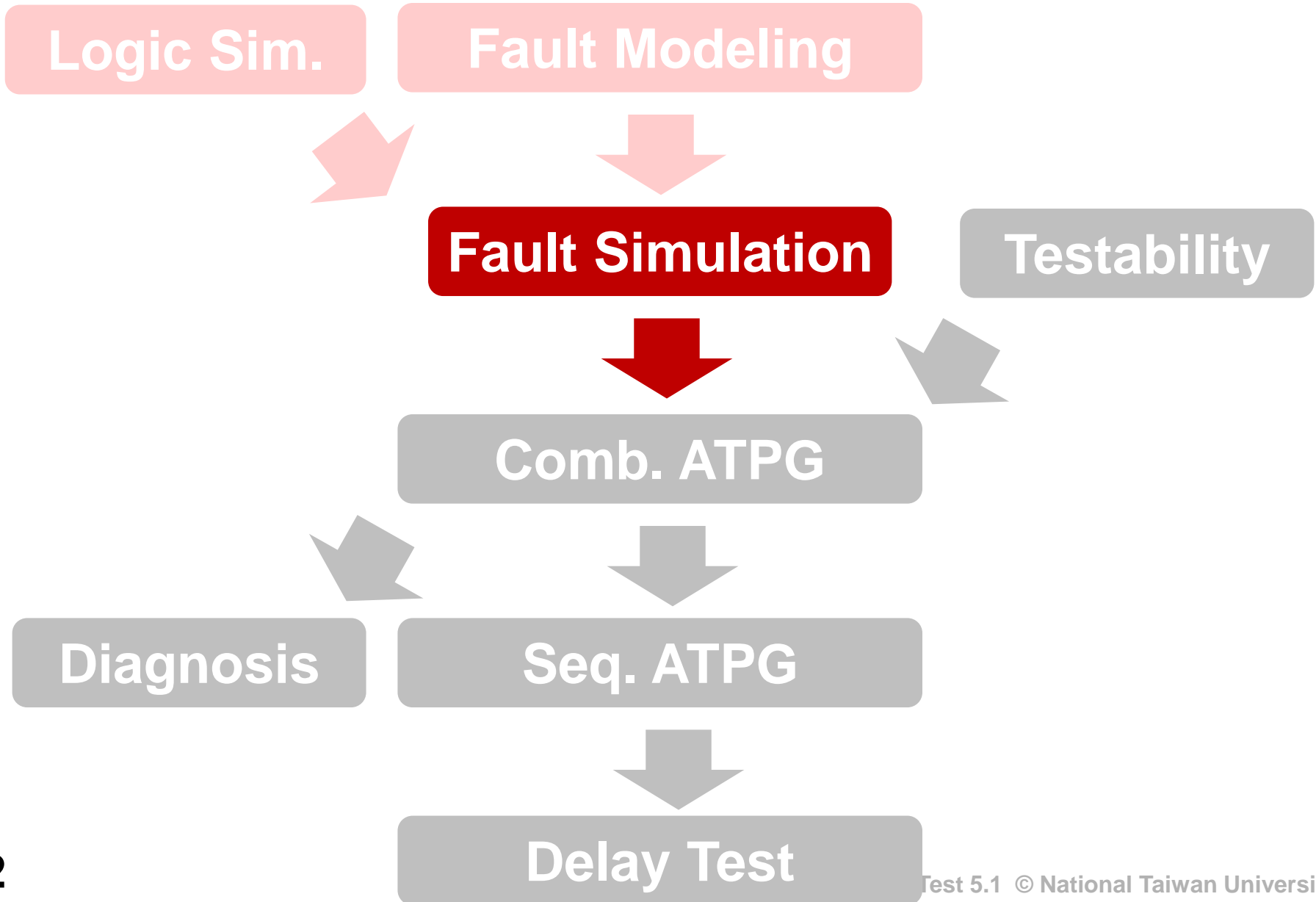# VLSI Testing
積體電路測試

Fault Simulation

Professor James Chien-Mo Li 李建模

Lab. of Dependable Systems (LaDS)

**Graduate Institute of Electronics Engineering
National Taiwan University**

1

Course Roadmap (EDA Topics)

Logic Sim.   Fault Modeling

Fault Simulation   Testability

Comb. ATPG

Diagnosis   Seq. ATPG

Delay Test

Test 5.1 © National Taiwan University

Motivating Problem

- **Apply 3 test patterns: P_1, P_2, P_3: {0,1,0} {0,0,1} {1,0,0}**
  - ♦ **Your manager asks you :** **What is fault coverage?**
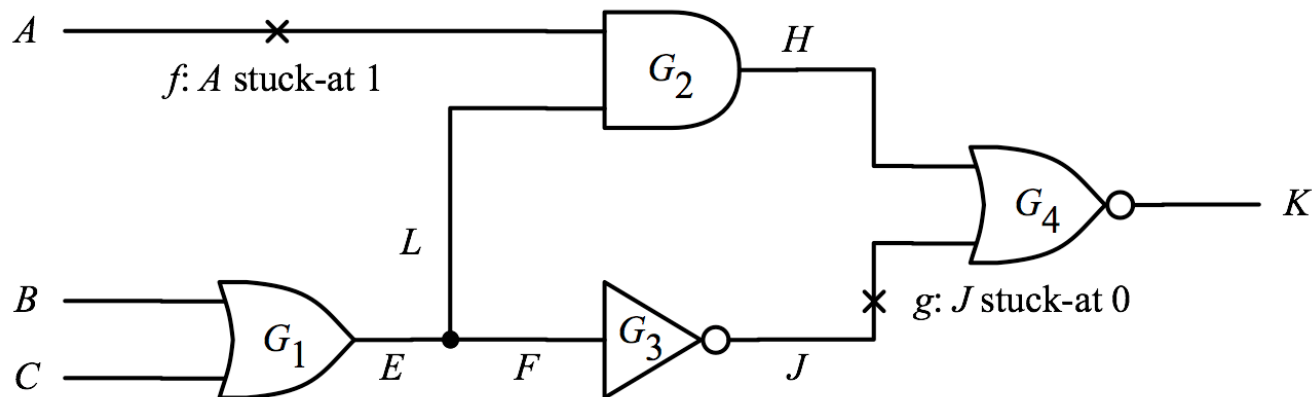
Pat.	Input			Internal					Output
	A	*B*	*C*	*E*	*F*	*L*	*J*	*H*	*K*
P_1	0	1	0	1	1	1	0	0	1
P_2	0	0	1	1	1	1	0	0	1
P_3	1	0	0	0	0	0	1	0	0

3

# Why Am I Learning This?

- **Fault simulation can**
    - ♦ **1. Determine fault coverage**
    - ♦ **2. Guide ATPG**
    - ♦ **3. Diagnose failed circuits**

> *"One sees qualities at a distance and defects at close range."*
>
> *(Victor Hugo)*

4

Fault Simulation

- **Introduction**
- **Fault simulation techniques**
- **Comparison of fault simulation techniques**
- **Alternatives to fault simulation**
- **Issues of fault simulation**
- **Concluding remarks**

5
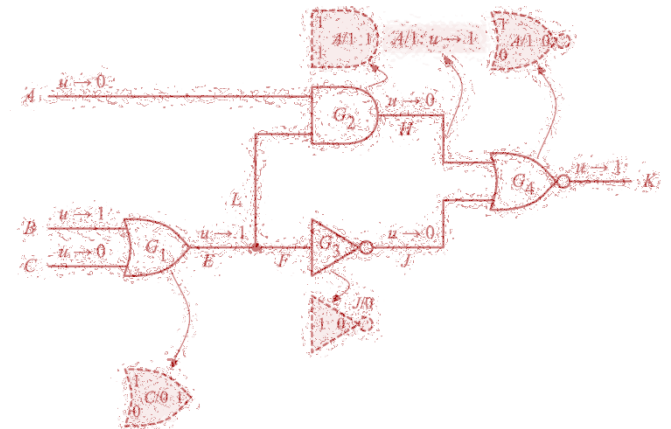
What Is Fault Simulation?

- **Given:**
  - ◆ **Circuit, fault model, test set**
- **Determine:**
  - ◆ **Output responses of faulty circuits**
  - ◆ **Detected faults, Undetected faults**
  - ◆ **Fault coverage**

```
                    ┌──────────────┐
                    │   Commands   │
                    └──────┬───────┘
                           │
                           ▼
┌──────────────┐    ┌──────────────┐    ┌──────────────┐
│ Input Stimuli│───▶│    Fault     │───▶│ Good/Faulty  │
│  (test set)  │    │  Simulator   │    │  Responses   │
└──────────────┘    └──────▲───────┘    └──────┬───────┘
                           │                   │
                           │                   ▼
                    ┌──────────────┐    ┌──────────────┐
                    │    Models    │    │Fault coverage│
                    │(circuit, fault)   │(Un)detected  │
                    └──────────────┘    │   faults     │
                                        └──────────────┘
```
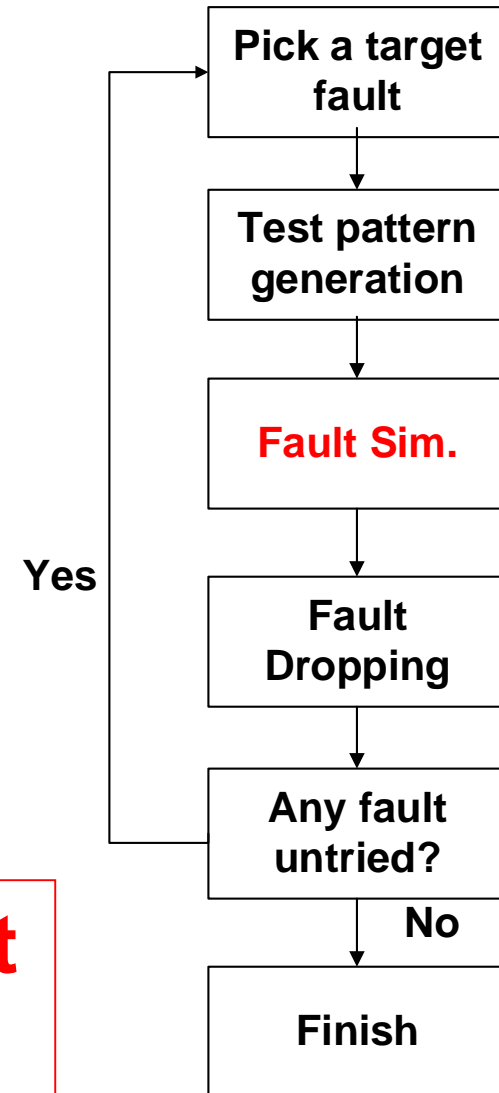
6
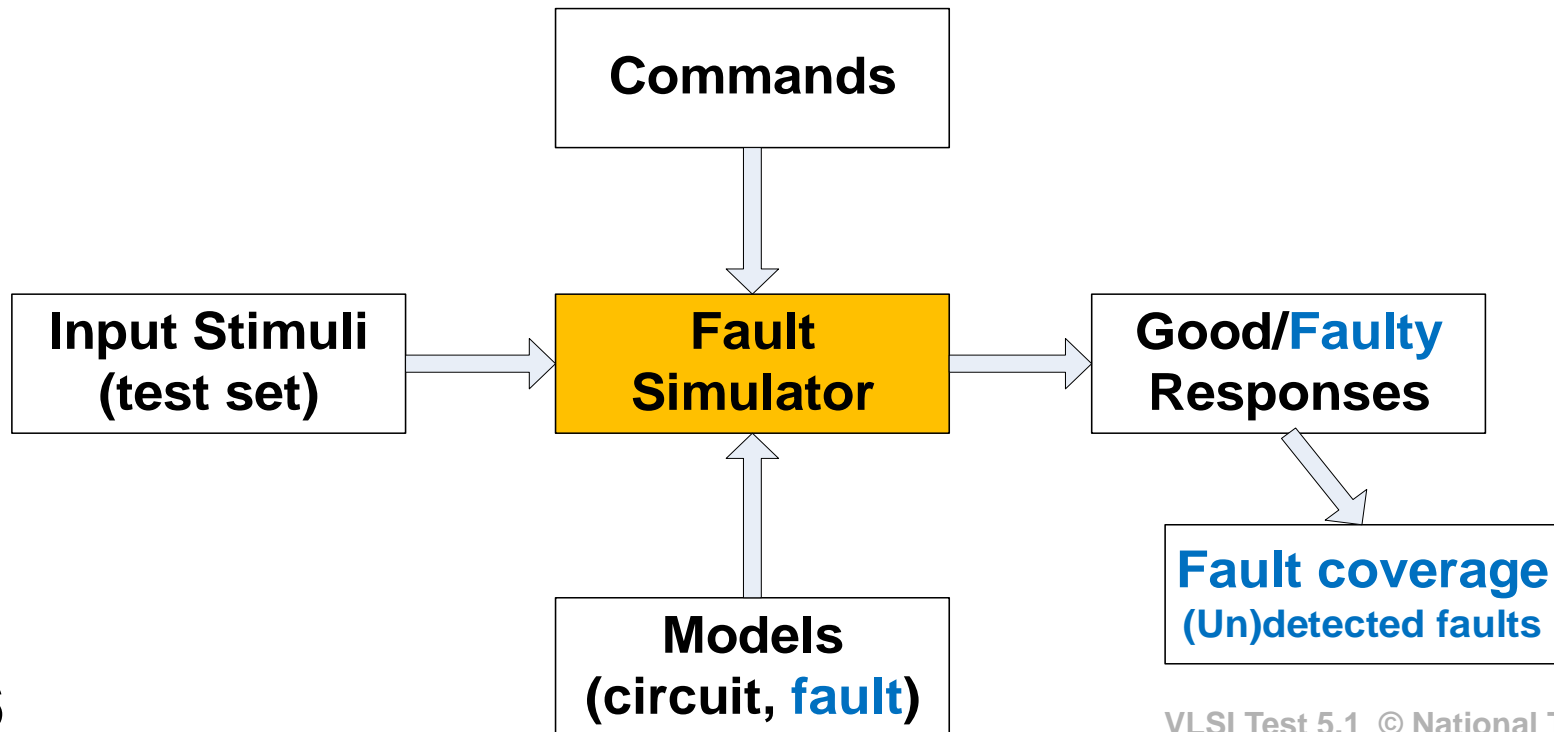
Applications of Fault Simulation

- **1. Evaluate quality of test sets (aka. *Fault Grading*)**
  - ◆ **Determine fault coverage of a test set**
- **2. Automatic Test Pattern Generation (ATPG)**
  - ◆ **Identify detected faults and undetected faults**
- **3. *Diagnosis* (See diagnosis chapter)**
  - ◆ **Generate *fault dictionary***
  - ◆ **Find culprit fault responsible for failure**

- **Therefore, fault simulators can be used as either**
  - ◆ **Stand alone tool, or**
    - ∗ **Fault grading**
  - ◆ **Embedded tool**
    - ∗ **ATPG**
    - ∗ **Diagnosis**

7

Fault Sim. Embedded in ATPG

- **After each test pattern generation**
  - ♦ **fault simulation**
- ***Fault dropping***
  - ♦ **Remove detected faults from fault list**
  - ♦ **Prevent repeated test generation for detected faults**
 - ♦ No dropping for diagnosis (see diagnosis chapter)

Fault Sim. Very Important for ATPG

Pick a target fault

↓

Test pattern generation

↓

Fault Sim.

↓

Yes

Fault Dropping

↓

Any fault untried?

No

↓

Finish

Fault Simulation

- **Introduction**
- **Fault simulation techniques**
  - ◆ **Serial fault simulation**
  - ◆ **Parallel fault simulation (1965)**
  - ◆ **PPSFP (1985)**
  - ◆ **Deductive fault simulation (1972)**
  - ◆ **Concurrent fault simulation (1974)**
  - ◆ **Differential fault simulation (1989)**
- **Alternatives to fault simulation**
- **Issues of fault simulation**
- **Concluding remarks**

9
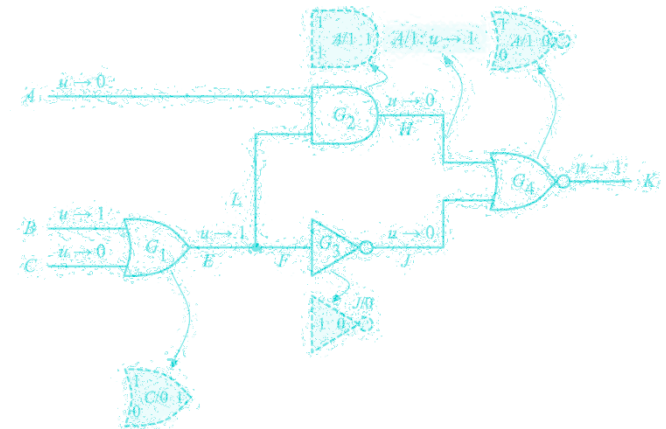
Serial Fault Simulation

- **Simple Idea: Run a fault-free logic simulation, store good outputs**
  - ♦ **For every fault**
    - ∗ **Modify good circuit (*fault injection*) to obtain a faulty circuit**
    - ∗ **Run logic simulation on faulty circuit**
    - ∗ **Compare faulty outputs with stored good outputs**
      - − **fault is detected if they are different**
- **Advantages**
  - ♦ **Easy to implement (regular logic simulator).**
  - ♦ **Ability to simulate many fault models (stuck-at, delay, Br, …)**
- **Disadvantage**
  - ♦ **Long CPU time**

10

Example

- **Consider only 2 faults: *f , g***   **Assume no fault dropping**
- **Given 3 patterns: P_1, P_2, P_3**
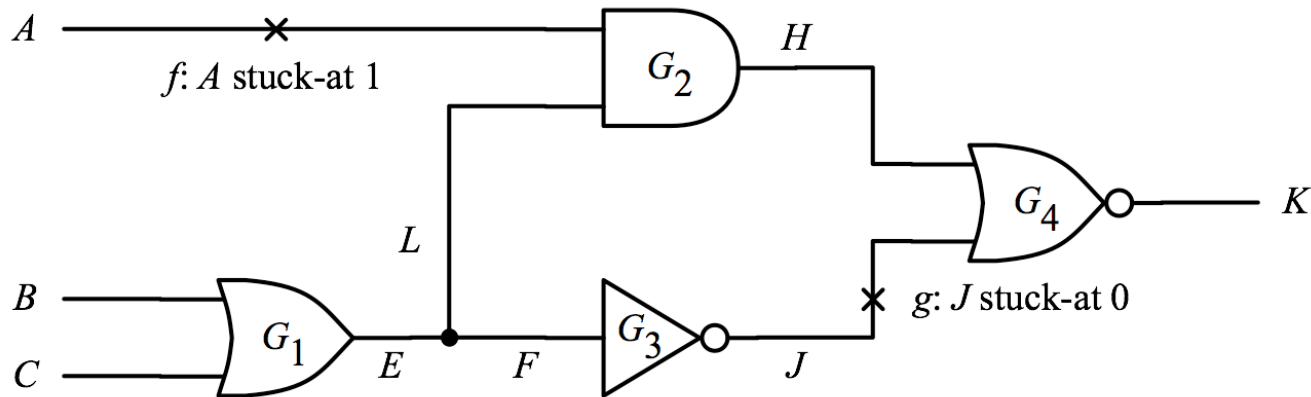
Pat. #	Input			Internal					Output		
	A	*B*	*C*	*E*	*F*	*L*	*J*	*H*	K_{good}	K_f	K_g
P_1	0	1	0	1	1	1	0	0	1	<u>0</u>	1
P_2	0	0	1	1	1	1	0	0	1	<u>0</u>	1
P_3	1	0	0	0	0	0	1	0	0	0	<u>1</u>

underline = detection

Serial Fault Simulation Flow

- **Inner loop: patterns**
- **Outer loop: faults**
  - ♦ **Speedup with fault dropping**

(WWW Fig 3.23)

```
                    ( start )
                        │
                        ▼
        F ← collapsed fault list
                        │
                        ▼
        fault-free simulation for all
                  patterns
                        │
                        ▼●◄──────────────────────┐
              no ┌──────────┐                     │
   ( end )◄──────│ next fault? │                  │
              └──────────┘                        │
                        │ yes                      │
                        ▼                          │
        1. get next fault f from F                 │
        2. reset pattern counter                   │
                        │                          │
                        ▼●◄──────────────┐         │
              no ┌──────────┐            │         │
         ┌───────│ next pattern? │───────┼────────►●
         │       └──────────┘            │         │
         │            │ yes              │         │
         │            ▼                  │         │
         │  1. get next pattern p        │         │
         │  2. fault simulation for p    │         │
         │            │                  │         │
         │            ▼                  │         │
         │  no ┌──────────┐              │         │
         └─────│ mismatch? │             │         │
              └──────────┘              │         │
                        │ yes            │         │
                        ▼                │         │
            delete f from F ─────────────┴─────────┘
```
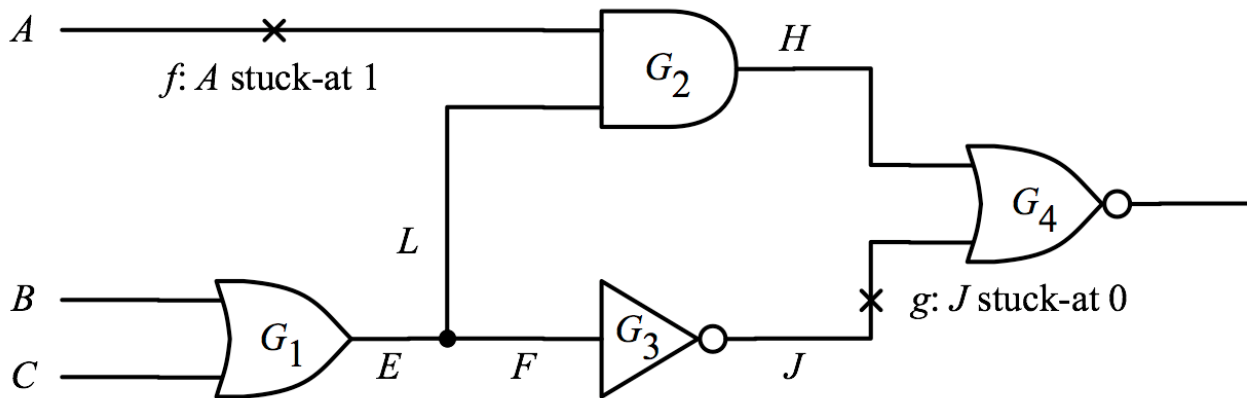
12
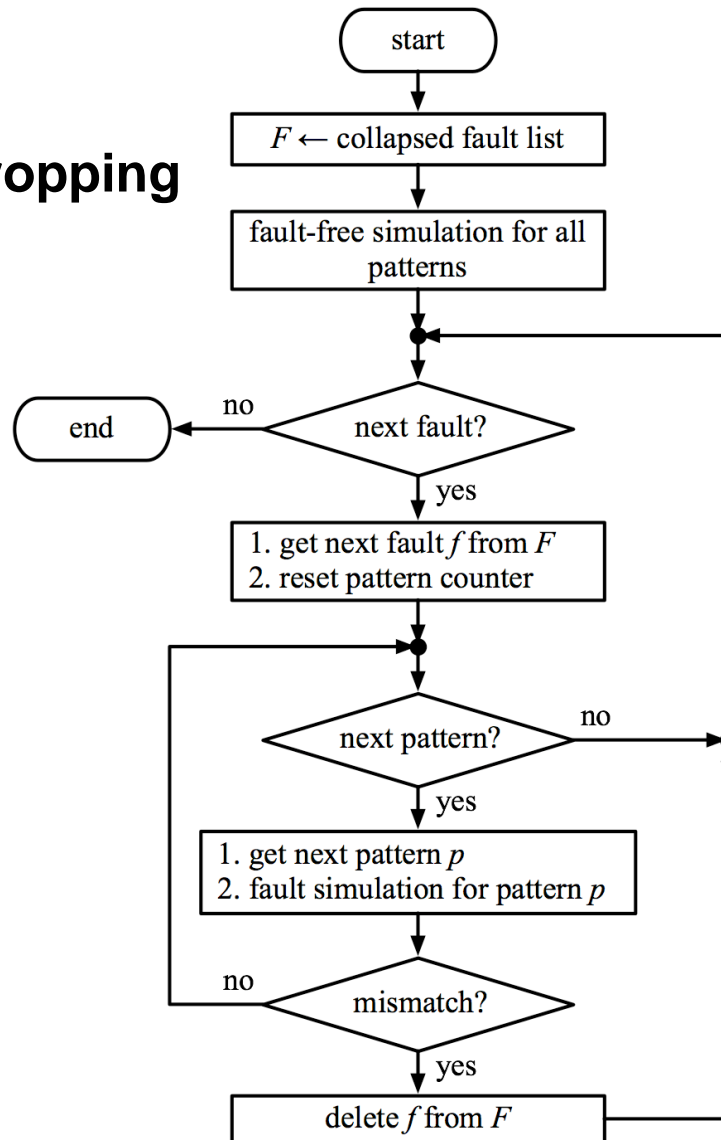
Quiz

Q: Apply 3 test patterns: P_1, P_2, P_3: {0,1,0} {0,0,1} {1,0,0}. Please use serial fault simulation to determine fault coverage =? Consider all 18 faults.

A:

f: A stuck-at 1

g: J stuck-at 0

Pat.	Input			Internal					Output
	A	**B**	**C**	**E**	**F**	**L**	**J**	**H**	**K**
P_1	0	1	0	1	1	1	0	0	1
P_2	0	0	1	1	1	1	0	0	1
P_3	1	0	0	0	0	0	1	0	0

fault	Det. by
A/1	P_1
A/0	-
B/1	
B/0	
C/1	
C/0	
E/1	
E/0	
F/1	
F/0	
H/1	
H/0	
J/1	
J/0	
K/1	
K/0	
L/1	
L/0	

13

Complexity of Fault Simulation

- **Fault Simulation Complexity** $O(FxPxG)$
  - ◆ *F*: number of faults
  - ◆ *P*: number of test patterns
  - ◆ *G*: number of gates
- **Comparison**
  - **Logic simulation:** $O(G \times P)$
  - ◆ **ATPG:** $O(G \times 2^{number\_of\_PI} \times F)$

- **So, use fault simulation to guide ATPG makes sense**
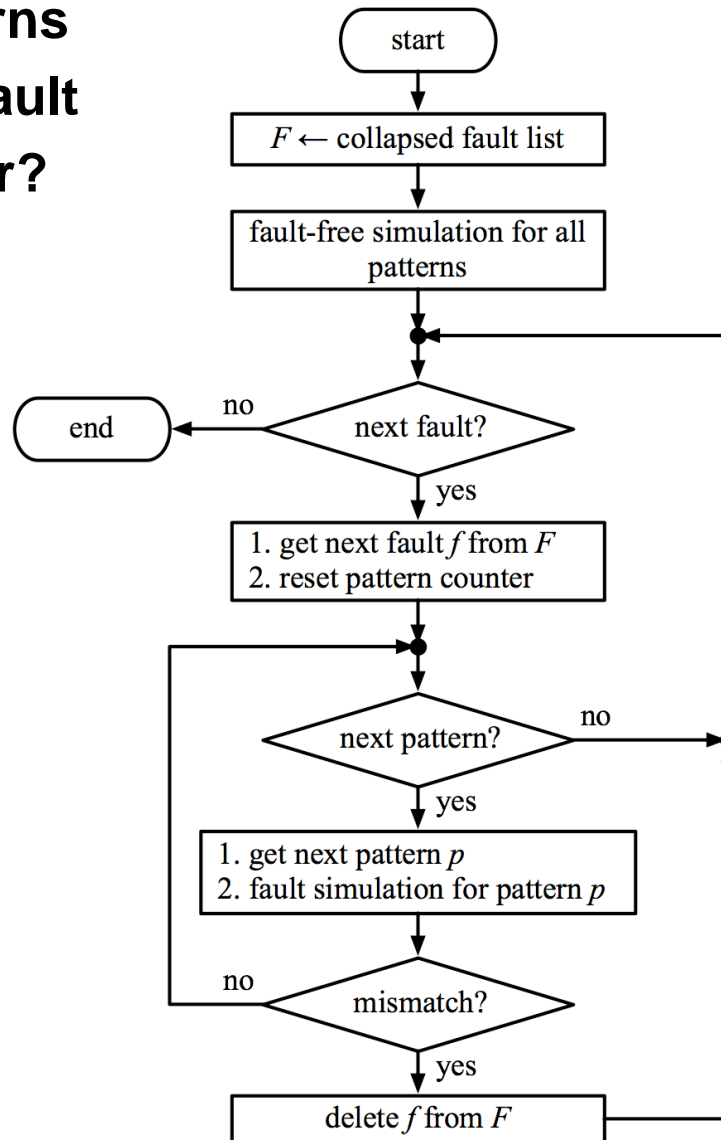
ATPG > Fault Sim > Logic Sim

Fault Simulation

- **Introduction**
  - **Fault simulation produces faulty circuit responses**
  - **Application: fault grading, ATPG, diagnosis**
  - **Fault simulation is polynomial time O(PxGxF)**
- **Serial fault simulation**
  - **Simulate one fault by one fault**

15
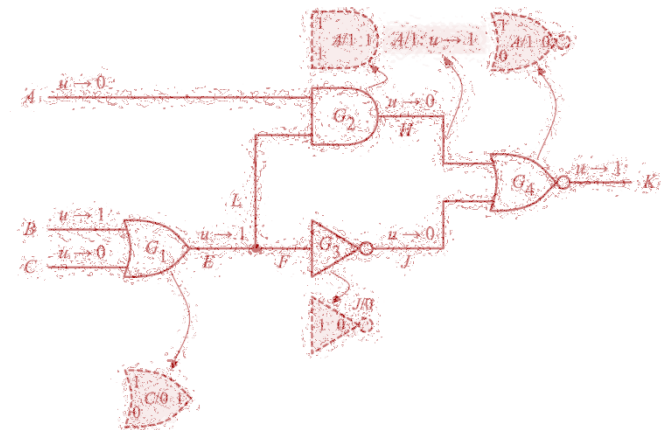
FFT

- **Q: why inner loop=patterns**
  - ♦ **why not inner loop=fault**
  - ♦ **So we can drop faster?**

(WWW Fig 3.23)

start

$F \leftarrow$ collapsed fault list

fault-free simulation for all patterns

next fault? — no → end

yes

1. get next fault f from F
2. reset pattern counter

next pattern? — no →

yes

1. get next pattern p
2. fault simulation for pattern p

mismatch? — no →

yes

delete f from F