PLA Test, Delay Test & STG Test 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, fault 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 Modeling

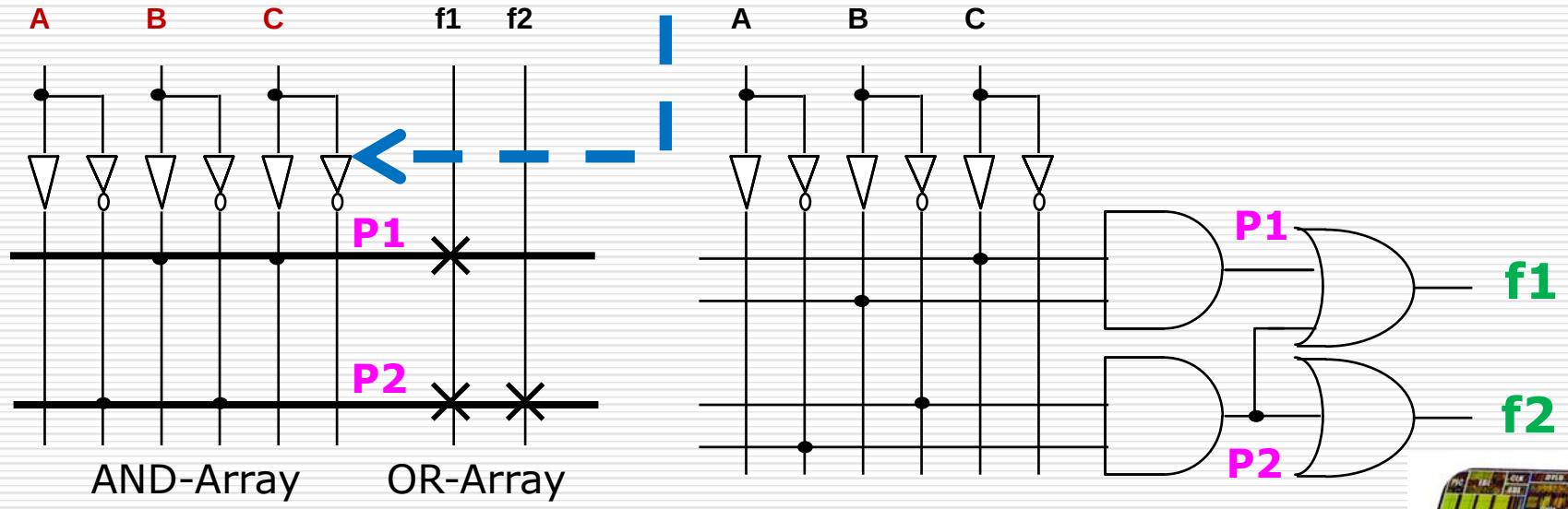
- **Fault Models**
  - **Stuck-At Faults**
  - **Bridging Faults**
  - **Transistor Stuck-On/Open Faults**
  - **IDDQ Faults**
  - **Functional Faults**
  - **Memory Faults**
  - **PLA Faults**
  - **Delay Faults & State Transition Faults**

PLA Faults (1/6)

- Stuck Faults
- Crosspoint Faults
  - Extra/Missing Transistors
- Bridging Faults
- Break Faults

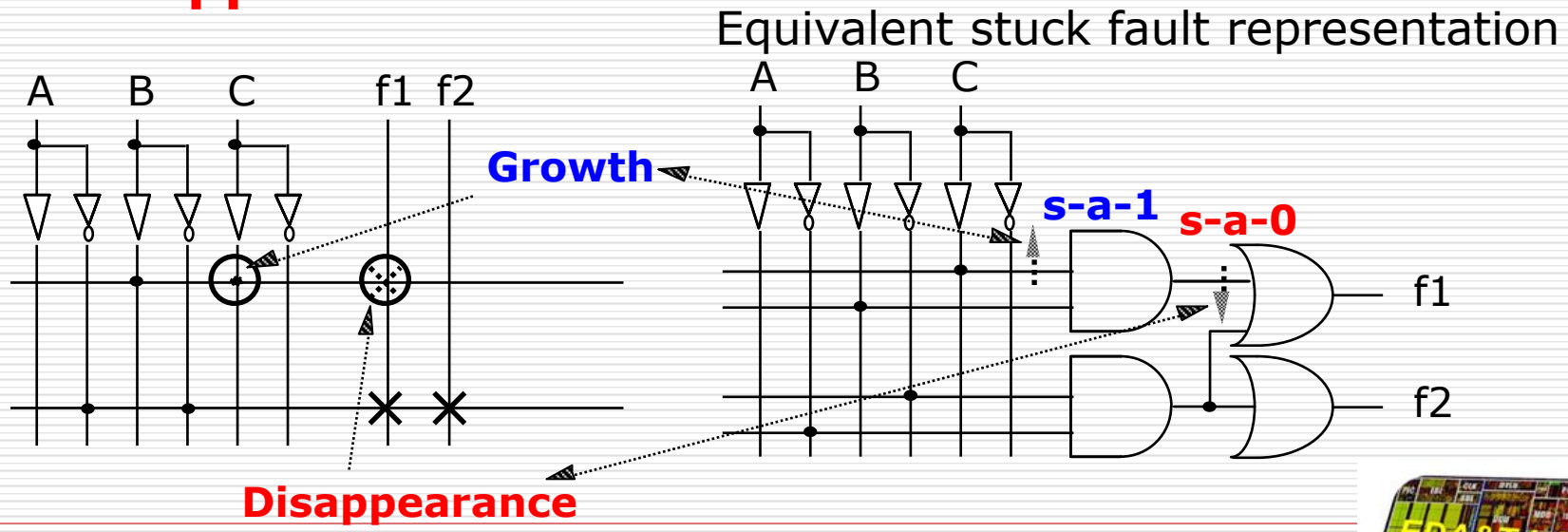
PLA Faults (2/6)

- Stuck-at faults
  - S-A-0 & S-A-1 on **inputs**, **input invertors**, **product lines**, and **outputs**
  - Easy to simulate in gate model



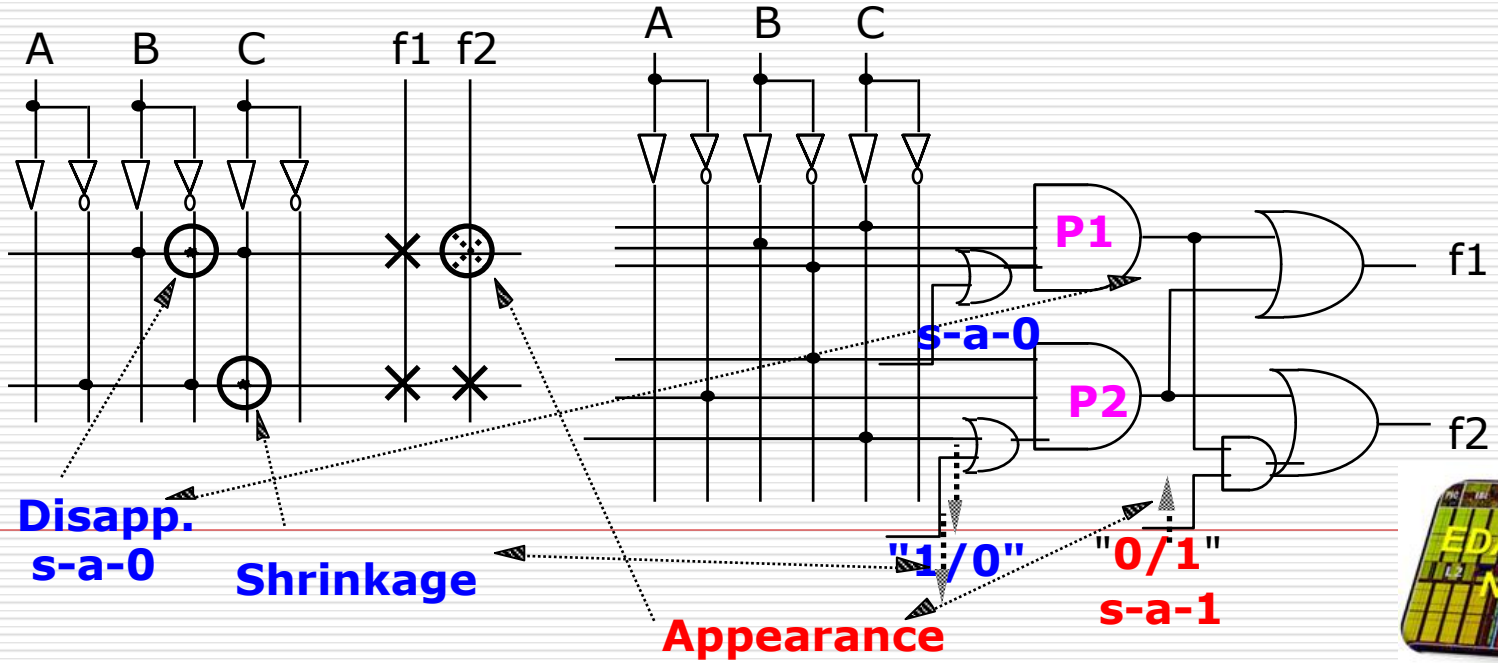
PLA Faults (3/6)

- Missing Crosspoint Faults
  - **Missing crosspoint in AND-array**
    - **Growth Fault**
  - **Missing crosspoint in OR-array**
    - **Disappearance Fault**



PLA Faults (4/6)

- Extra Crosspoint Faults
  - **Extra crosspoint in AND-array**
    - **Shrinkage or Disappearance Fault (B&Bbar)**
  - **Extra crosspoint in OR-array**
    - **Appearance Fault**



PLA Faults (5/6)

- Bridging faults
  - **Shorting of adjacent lines (layout dependent)**
  - **Faulty value identical on shorted lines**
  - **Faulty value AND/OR function of shorted signals**
  - **A large number of bridging faults map into stuck or crosspoint faults**

PLA Faults (6/6)

- Summary
  - **Crosspoint Faults**
    - **80 ~ 85% covered by stuck-fault tests**
    - **Layout-dependence in folded PLA**
  - **Bridging Faults**
    - **99% covered by stuck-fault tests**
    - **Layout-dependence in *all* PLA**
      - (Ref: Agrawal & Johnson, ICCD-86)

Fault Modeling

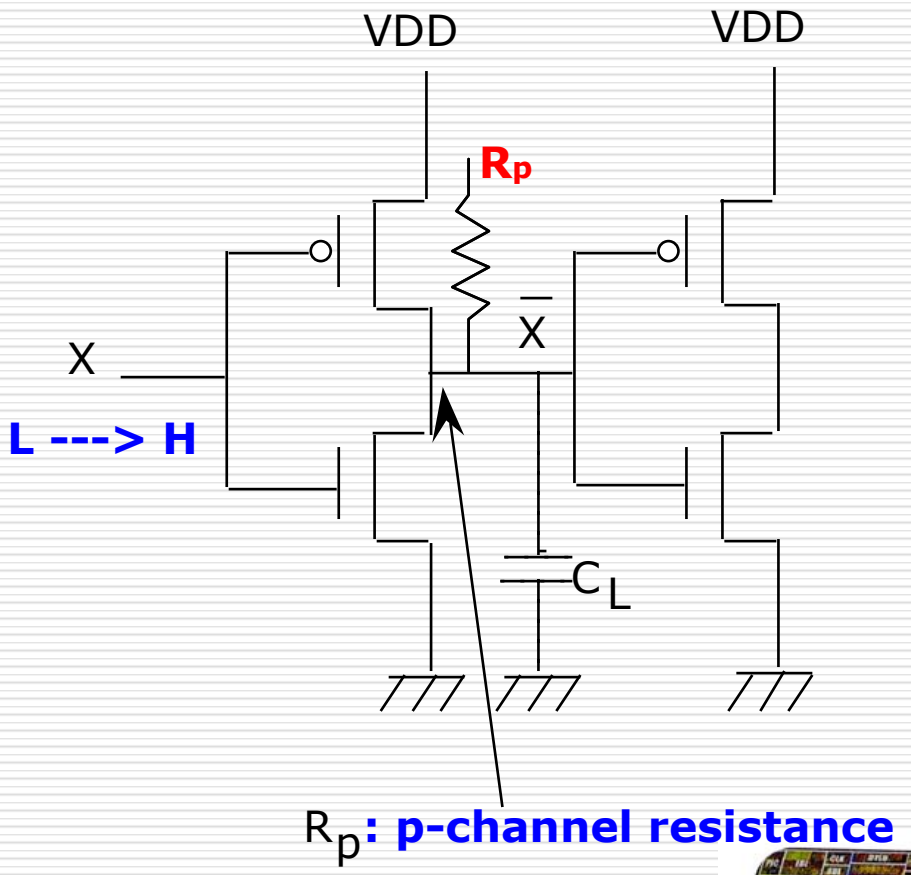
- **Fault Models**
  - **Stuck-At Faults**
  - **Bridging Faults**
  - **Transistor Stuck-On/Open Faults**
  - **IDDQ Faults**
  - **Functional Faults**
  - **Memory Faults**
  - **PLA Faults**
  - **Delay Faults & State Transition Faults**
    - **GATE Delay Faults**
    - **PATH Delay Faults**

Delay Faults (1)

- Why delay testing?
  - There are defects on the chip which allows it to pass the **DC stuck-fault testing**, but causes it to **fail when operated at system speed**
    - A chip may **pass testing under 1MHz operation but not under 10 MHz**
    - **Total delay in critical/longest paths should be less than/equal to 1/f**
      - **1MHz =>1/$(10)^6$ sec = $(10)^{-6}$ sec= 1 μs**
      - **10MHz =>1/$(10)^7$ sec = $(10)^{-7}$ sec= 0.1 μs**
      - **100MHz =>1/$(10)^8$ sec = $(10)^{-8}$ sec= 0.01 μs**
      - **1GHz =>1/$(10)^9$ sec = $(10)^{-9}$ sec= 1 *ns***

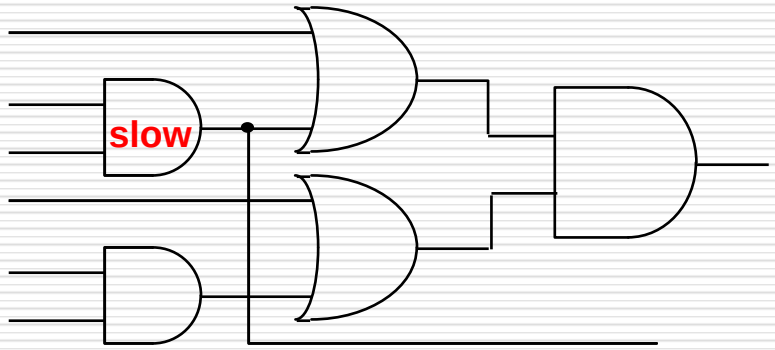
Delay Faults (2)

- **Gate**-Delay-Fault
  - **Slow** to rise, **slow** to fall
    - x: from L (0) => H (1)
    - $\overline{x}$ is **slow to rise**
      - when **channel resistance R_p is abnormally high**



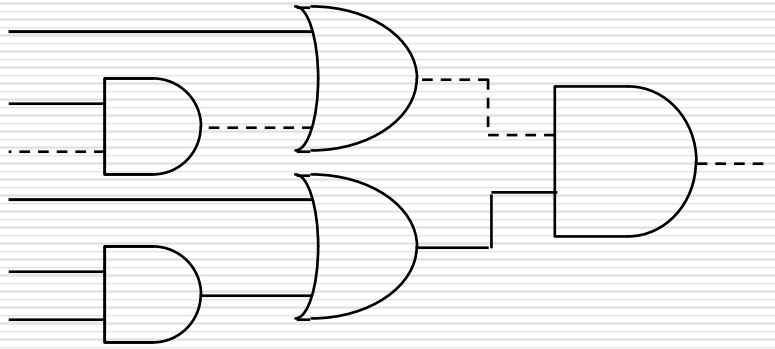
Delay Faults (3)

- **Gate**-Delay-Fault
- Disadvantage:
  - Delay faults resulting from **the sum of several small incremental delay defects** may not be detected=>Path-Delay-Fault



Delay Faults (4)

- **Path**-Delay-Fault
  - Propagation delay of the path **exceeds** the **clock interval/clock period**
  - The **number of paths** grows exponentially with the number of gates



Delay Faults (5)

- **Path**-Delay-Fault
  - Propagation delay of the path **exceeds** the **clock interval/clock period**
  - Slack = clock period – the delay of the maximum-delay path
  - **Critical path**: maximum-delay path
    - Path 1

https://www.edn.com/design/test-and-measurement/4381761/Small-delay-defect-testing-4381761

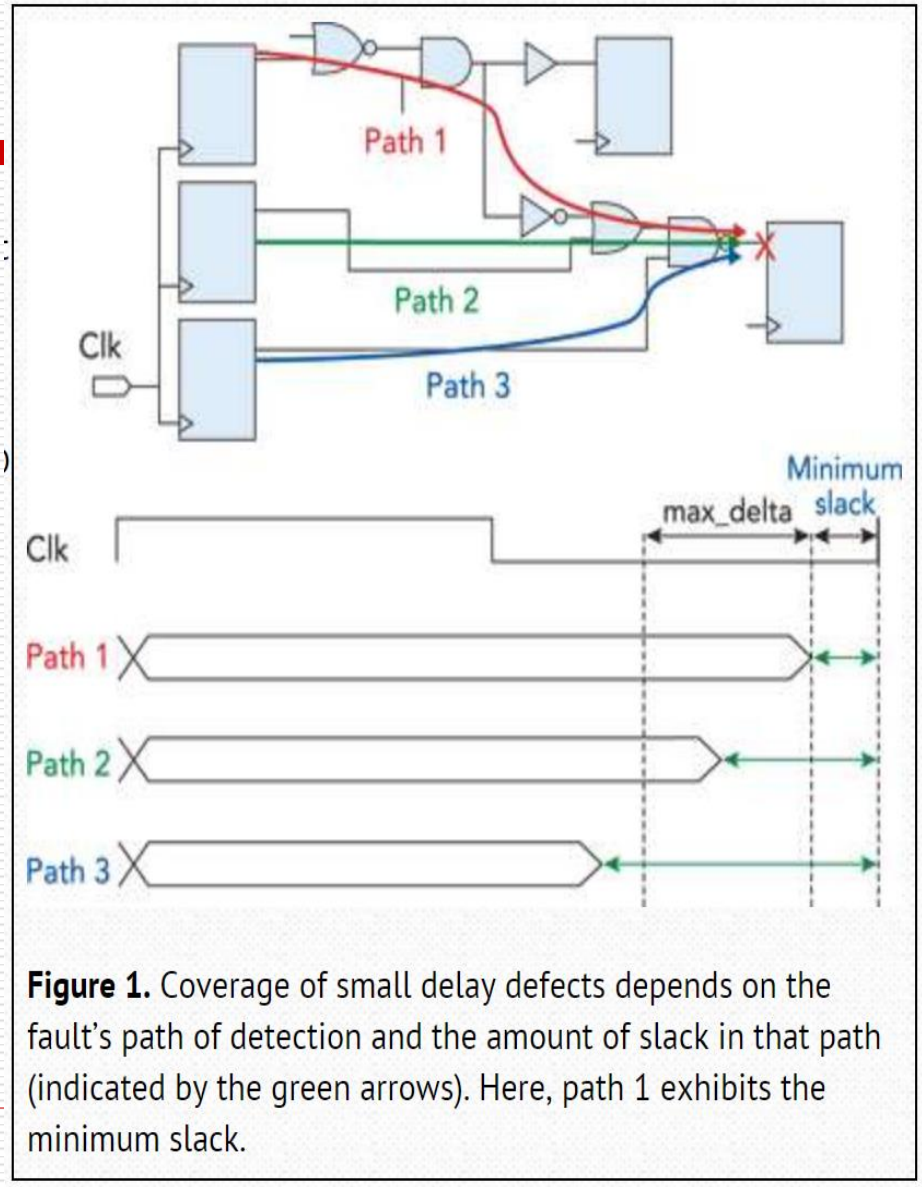


Figure 1. Coverage of small delay defects depends on the fault's path of detection and the amount of slack in that path (indicated by the green arrows). Here, path 1 exhibits the minimum slack.

Delay Faults (6)

- Semiconductor companies have come to rely on delay testing to attain high defect coverage of manufactured digital ICs
- **TD:** transition delay
  - TD ATPG improves defect coverage beyond the levels that stuck-at patterns alone can achieve
- **SDD**

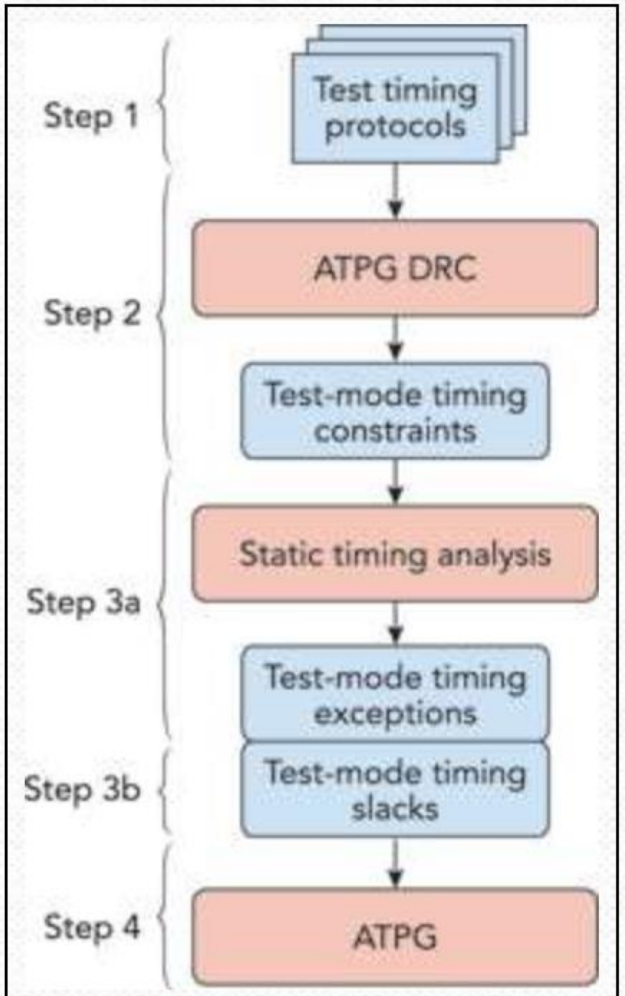


Figure 4. The flow for generating slack data and timing exception data for SDD ATPG involves a four-step process.

Delay Faults (7)

- **TD:** transition delay
  - +:
  - TD ATPG improves defect coverage beyond the levels that stuck-at patterns
  - -:

    TD ATPG methodology is limited in its ability to reach the test quality levels required for nanometer designs

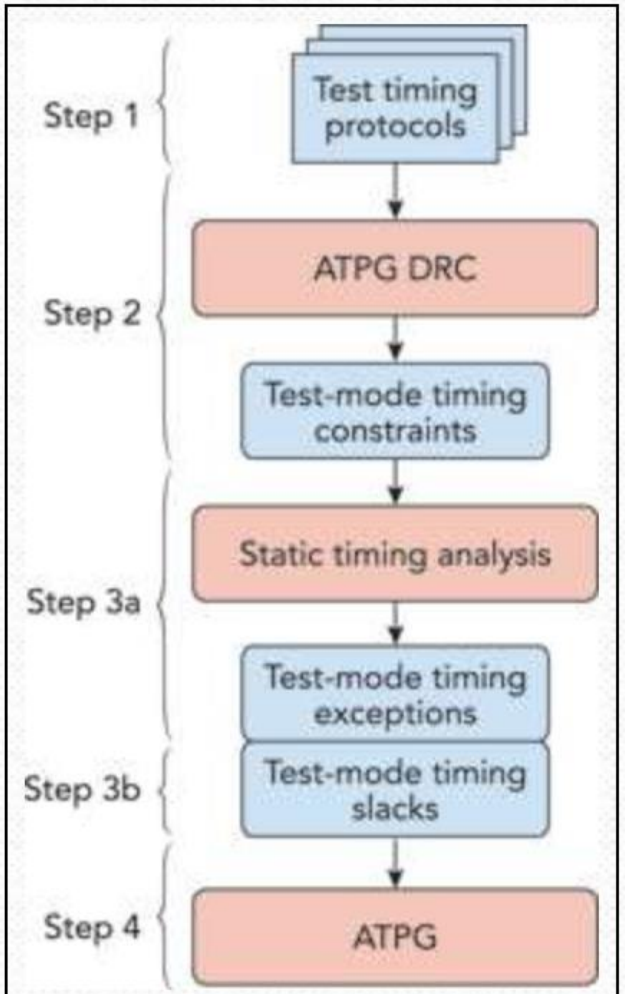


Figure 4. The flow for generating slack data and timing exception data for SDD ATPG involves a four-step process.

Delay Faults: SDD ATPG(8)

- **SDD:** Small Delay Defect
  - STMicroelectronics is deploying a new delay test methodology called **SDD (small delay defect) ATPG** as a means to achieve even higher defect coverage than standard TD ATPG

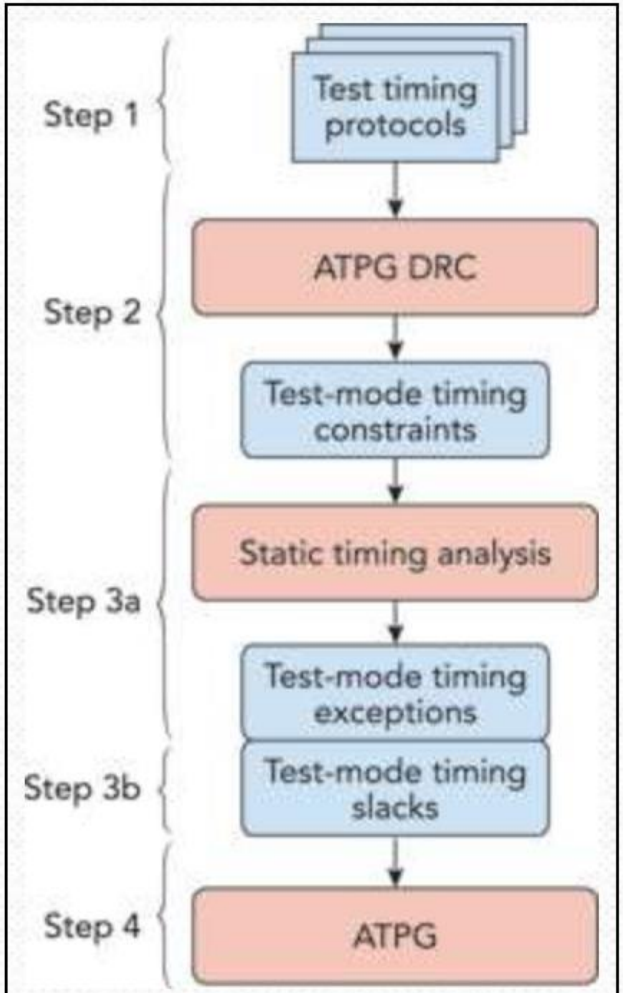


Figure 4. The flow for generating slack data and timing exception data for SDD ATPG involves a four-step process.

Delay Faults (9)

- "Delay defect" refers to any type of physical defect, or an interaction of defects, that adds enough signal-propagation delay in a device to produce an **invalid response** **when the device operates at the targeted frequency**

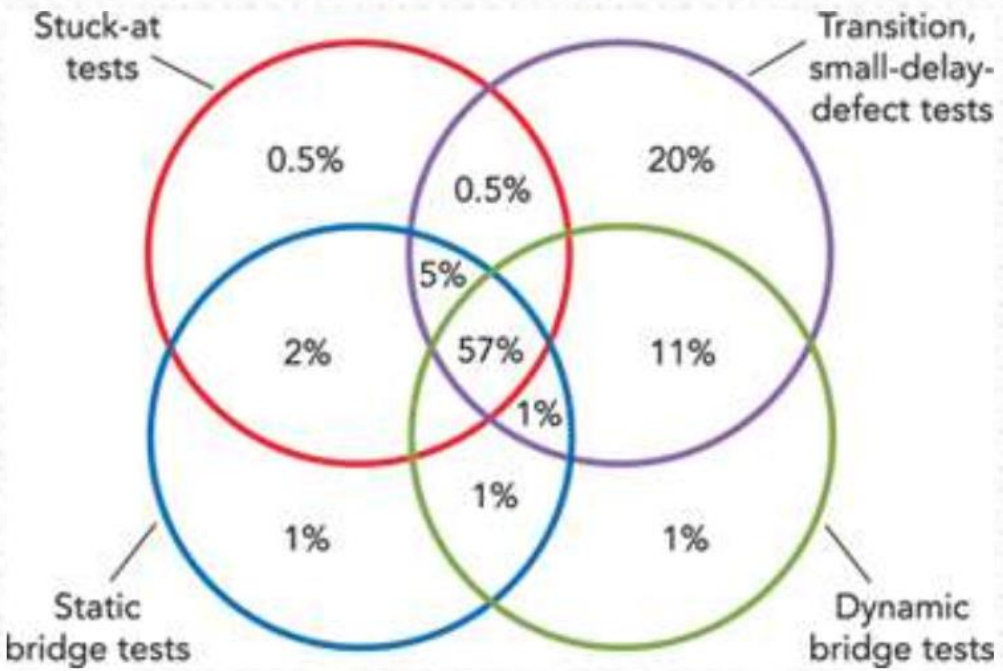
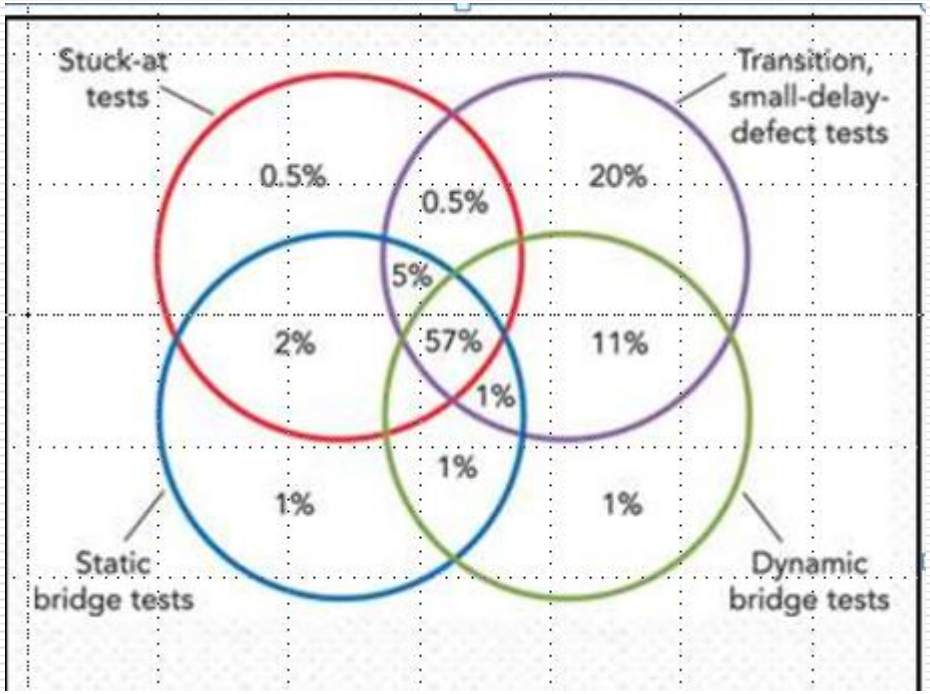


Figure 5. This Venn diagram shows the percentage of failing parts covered by each type of test.

Fault Model vs Test Coverage: Delay Faults vs. Bridge vs SA (10)

- SDD, TD: 94.5%
- Dynamic Bridge: 71%
- Static Bridge: 67%
- Stuck-at: 65%



Delay Faults : SDD (11)

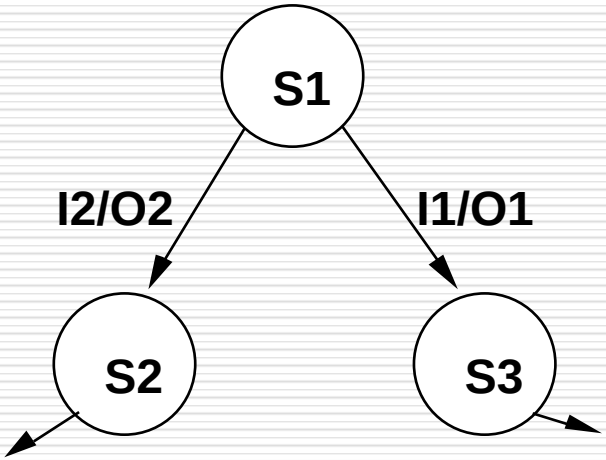
- majority of devices that fail due to delay defects fail because of "**small delay defects**"
- SDD contribute to delays much smaller than the clock cycle times associated with the process technology node
- Targeting these SDDs during test improves defect coverage and lowers the test escape rate, measured as **DPPM (defective parts per million)**

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**
    - **GATE Delay Faults**
    - **PATH Delay Faults**

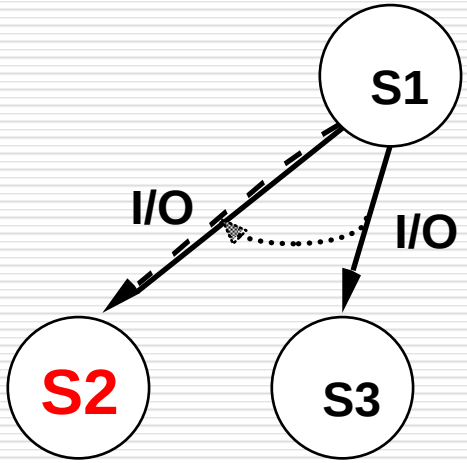
State Transition Graph (STG)

- State Transition Graph
  - **Mealy** Machine
    - Each state transition is associated with 4 tuple: (**source state, input, output, destination state**)



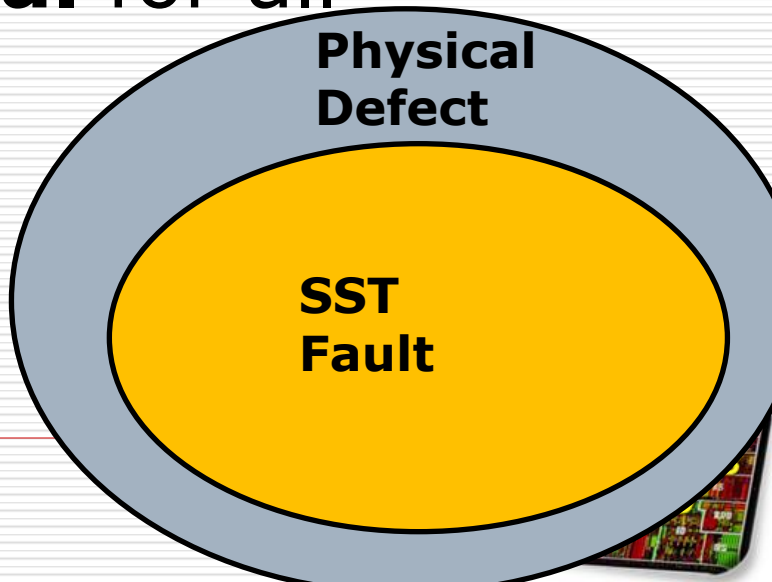
Single State Transition (SST) Fault Model (1/2)

- A fault causes a single state transition to a **wrong destination state**.



Single State Transition (SST) Fault Model (2/2)

- **M(N-1) faults** for a **M-transition N-state** machine
- **Modeled in the state transition level** and **independent of implementation**
- **Dominated by most physical** for all possible implementation

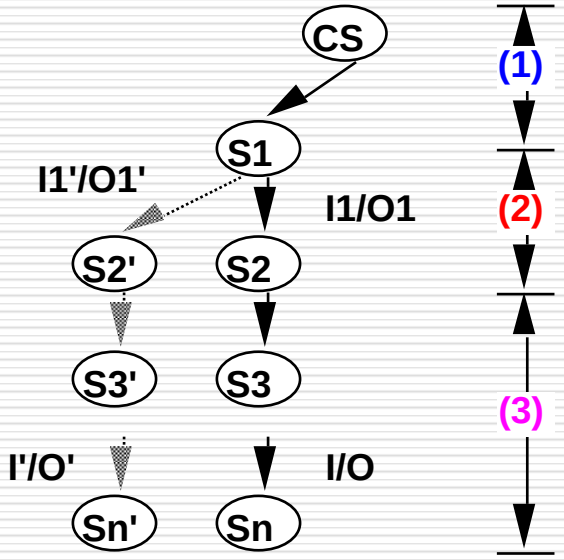


Transition Faults in State Graph

- Any **irredundant physical faults will cause some changes in the state graph**
  - Single or multiple transitions will be corrupted
  - A transition is corrupted if its:
    - **a) Output label is changed**
    - **b) Destination/Next State is changed**,
    - c) Both (a) and (b)
  - A test to type (b) fault will detect type (a) and (c) faults
  - 意思是說連看不到的Next State都可以被測試到，則Output (或Output&NextSate)都可被測試到

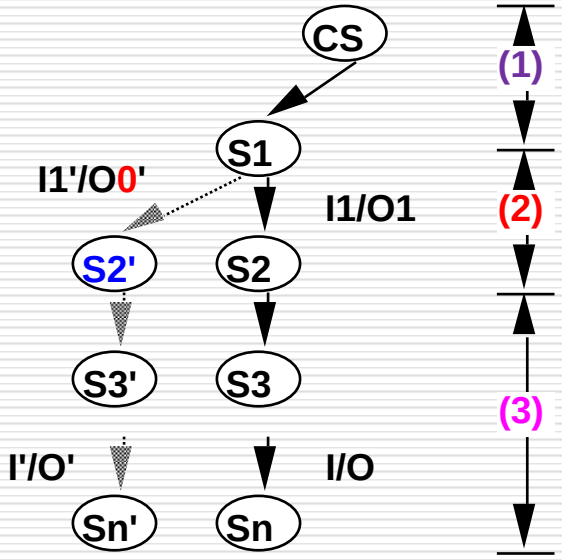
Test for a State Transition Fault (1/2)

- A **test sequence** for a fault causing a type (b) corruption on a transition consists of three subsequences:
  1) **Initialization sequence**
  2) **Input label** of the faulty transition
  3) **State-pair** differentiating sequence between good and faulty states



Test for a State Transition Fault (2/2)

- Subsequences (1) and (2) already produce **faulty output responses** if the output label of the target transition is corrupted
- Only need to generate **test inputs for the state transition faults** causing **wrong** destination states

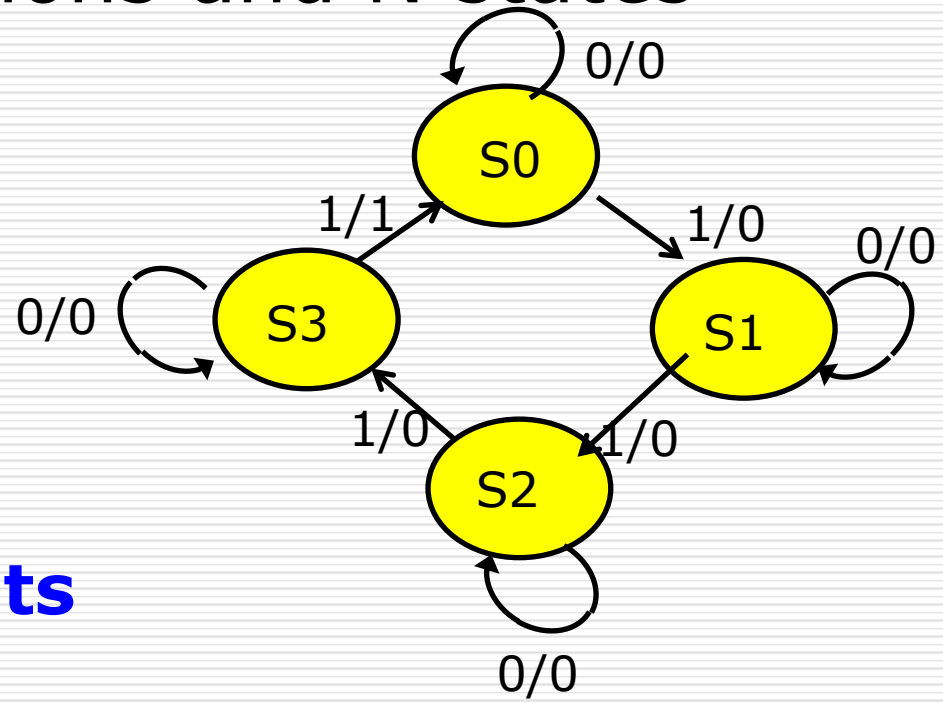


Multiple-State-Transition (MST) Faults

- A test sequence that detects all **MST** faults detects **all irredundant physical fault**
- A machine of **M** transitions and **N** states has:
  - **$N^M - 1$ MST faults**
  - **$M(N - 1)$ SST faults**
- **Summary:**
  - **Similar to: 2^{10} for #SSA for 10-inputs**
    for 0 and 1 two possibilities
  - **Thus, (#fixpossibilities)$^{\#variable}$**

Multiple vs. Single-State-Transition (MST) Faults

- A machine of M transitions and N states has:
  - **$N^M - 1$ MST faults**
  - **$M(N - 1)$ SST faults**
- 4-Counter:
  - **$4^{4x2} - 1$ MST faults**
  - **$(4x2)(4 - 1)$ SST faults**



Thank You for Your Attention & Participation!