Study on the Reliability Enhancement of Edge Computing Devices

Xihong Zhou

Supervisor: Hiroshi Takahashi (Full Professor)

Graduate School of Science and Engineering
Ehime University

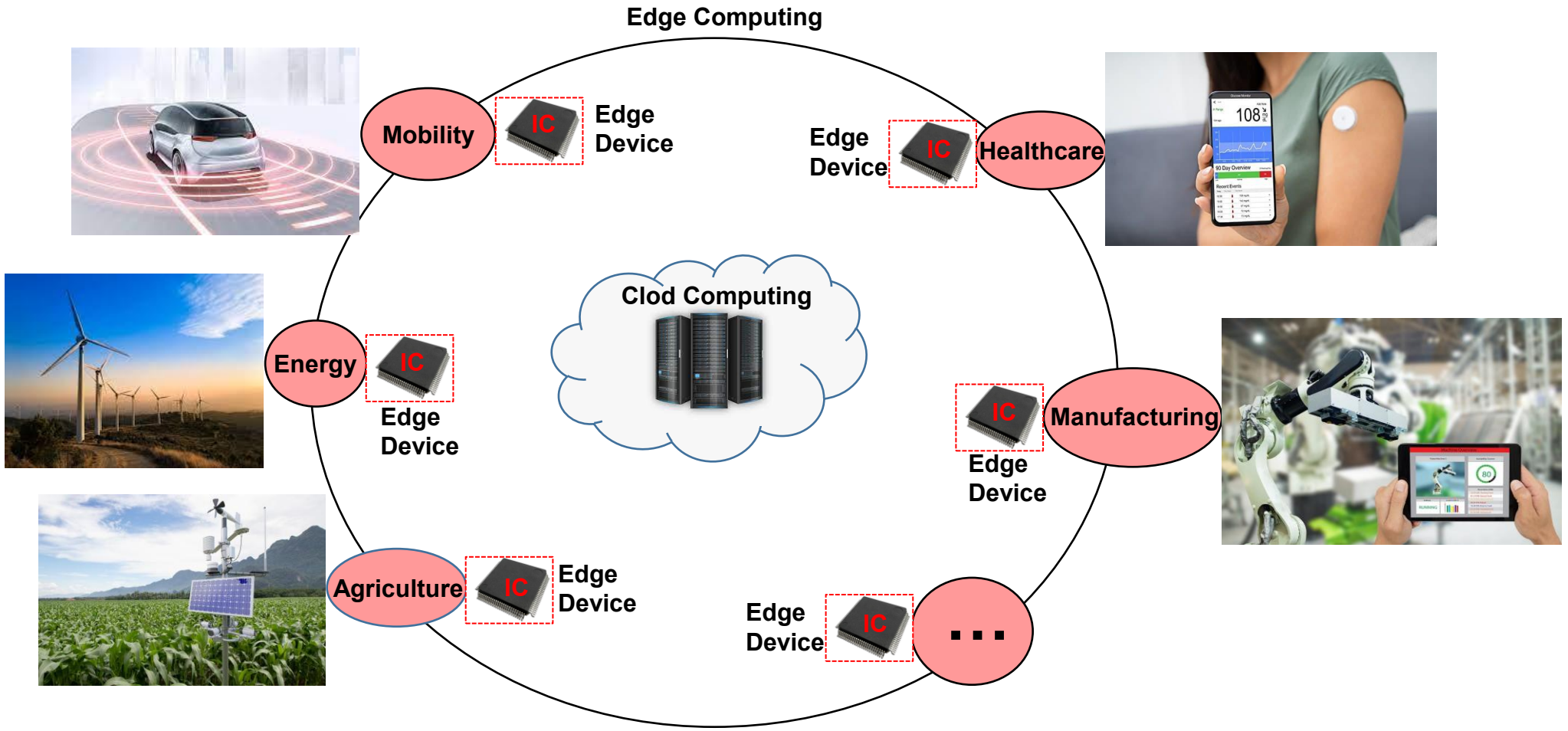
Doctor of Engineering (2023)

Outline

- Background
- Objectives
- Reliability Enhancement for Automotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Background ~ *Edge Devices* ~

- IC (Integrated Circuit) for Processing data at the edge of IoT/AI systems



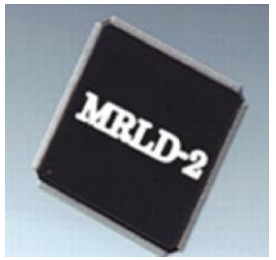
Reliability of edge devices must be ensured

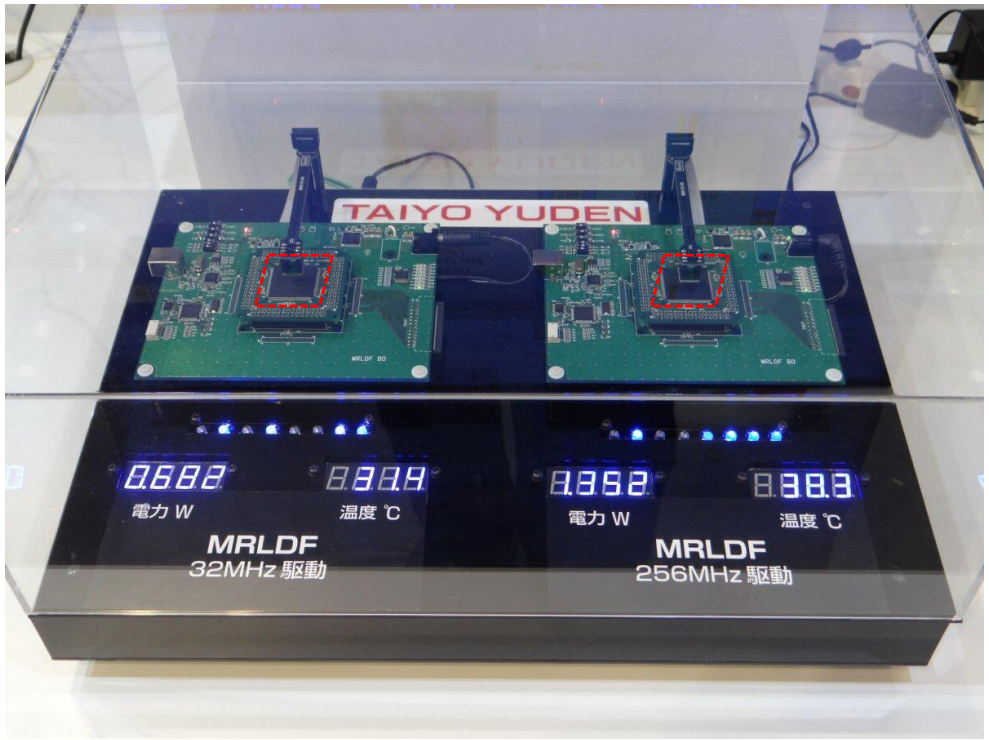
Objectives

- Enhancing the reliability of two distinct types of edge devices:
  - Automotive ECU (electronic control unit)
  - MPLD (memory-based programmable logic device)

ECU:

MPLD:



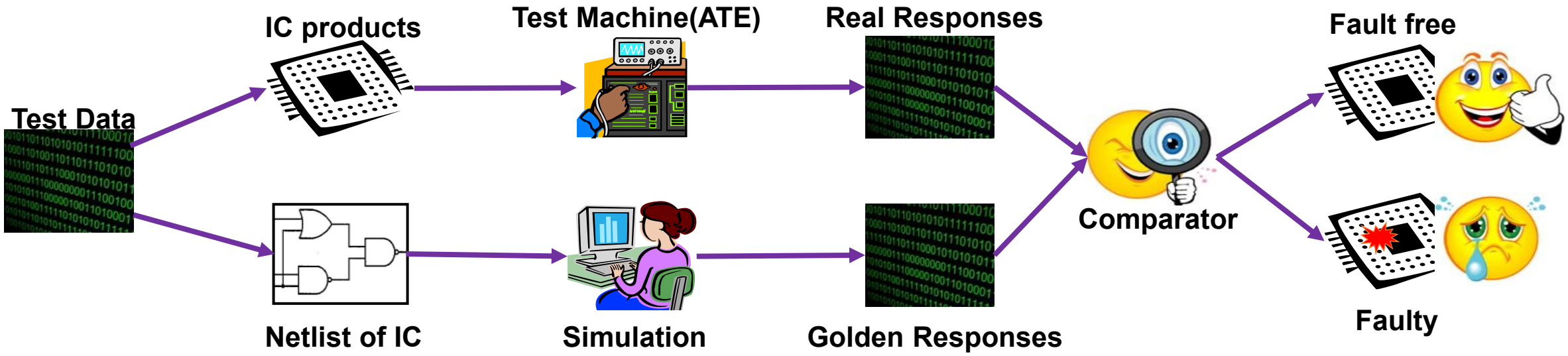


Outline

- Background
- Objectives
- **Reliability Enhancement for Automotive ECU**
  - **Multi-Cycle Power-on Self-Test (POST)**
  - **Test Point Insertion Technique for Multi-Cycle POST**
  - **Conclusions**
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Integrated Circuit (IC) Testing

- Defects will occur in Manufacturing Process or In-field Operation (Aging)
- Testing is a process to find whether an IC is Function correctly or Defective
  - Essential to improve the Yield and Reliability of ICs



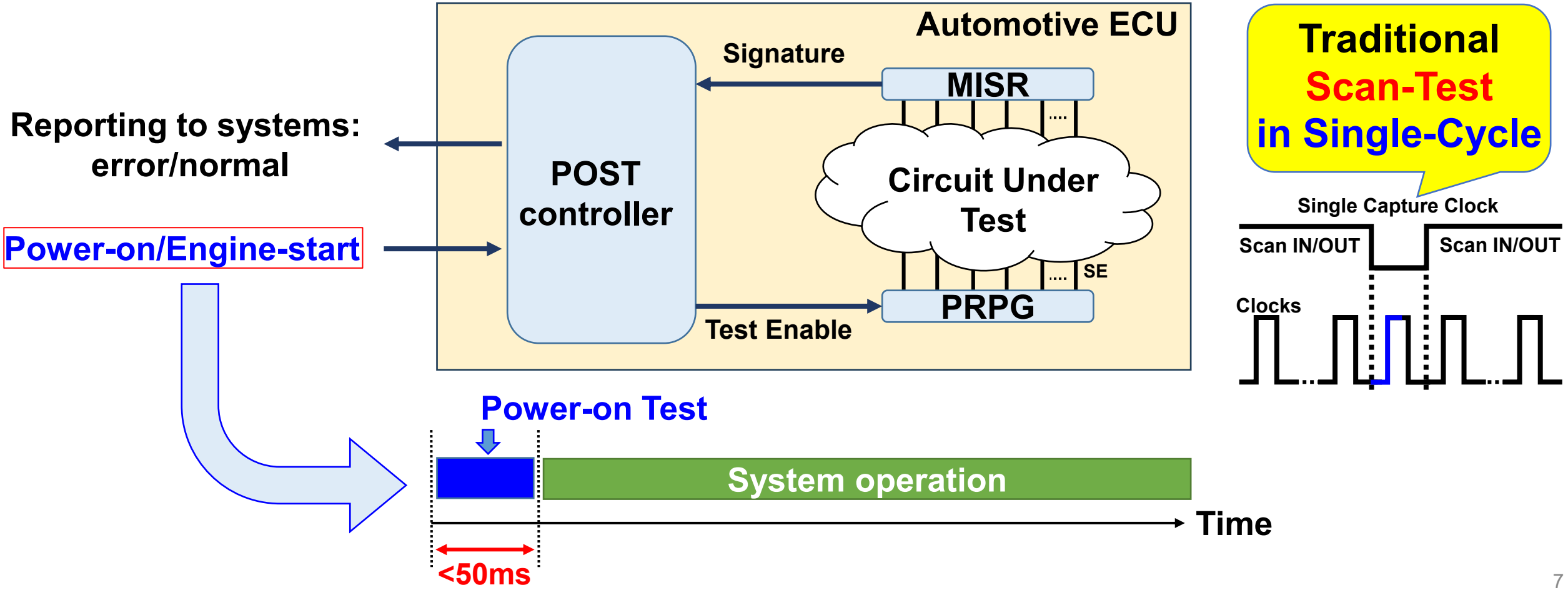
- Manufacturing Testing: testing ICs to ensure they are working as intended before shipment
  - Goal: identify and correct manufacturing defects(process variations, fabrication errors, etc.)

- In-Field Testing: testing during the actual use of ICs.
  - Goal: ensure the ongoing health and functionality of ICs under real-world condition

Functional Safety Requirement for ECU

- ISO 26262 Standard
  - ➢ Need for In-Field Testing
    - ✓ Power-On Self-Test (POST)

- ➢ Challenge to Achieve Efficient POST
  - ✓ Limited test application time (<50ms)
  - ✓ High fault coverage (>90%)

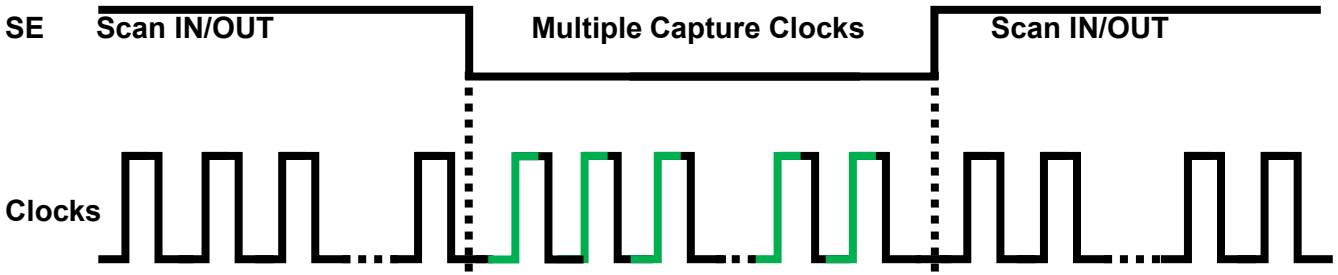


Outline

- Background
- Objectives
- **Reliability Enhancement for Automotive ECU**
  - **Multi-Cycle Power-on Self-Test (POST)**
  - **Test Point Insertion Technique for Multi-Cycle POST**
  - **Conclusions**
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Proposed Approach

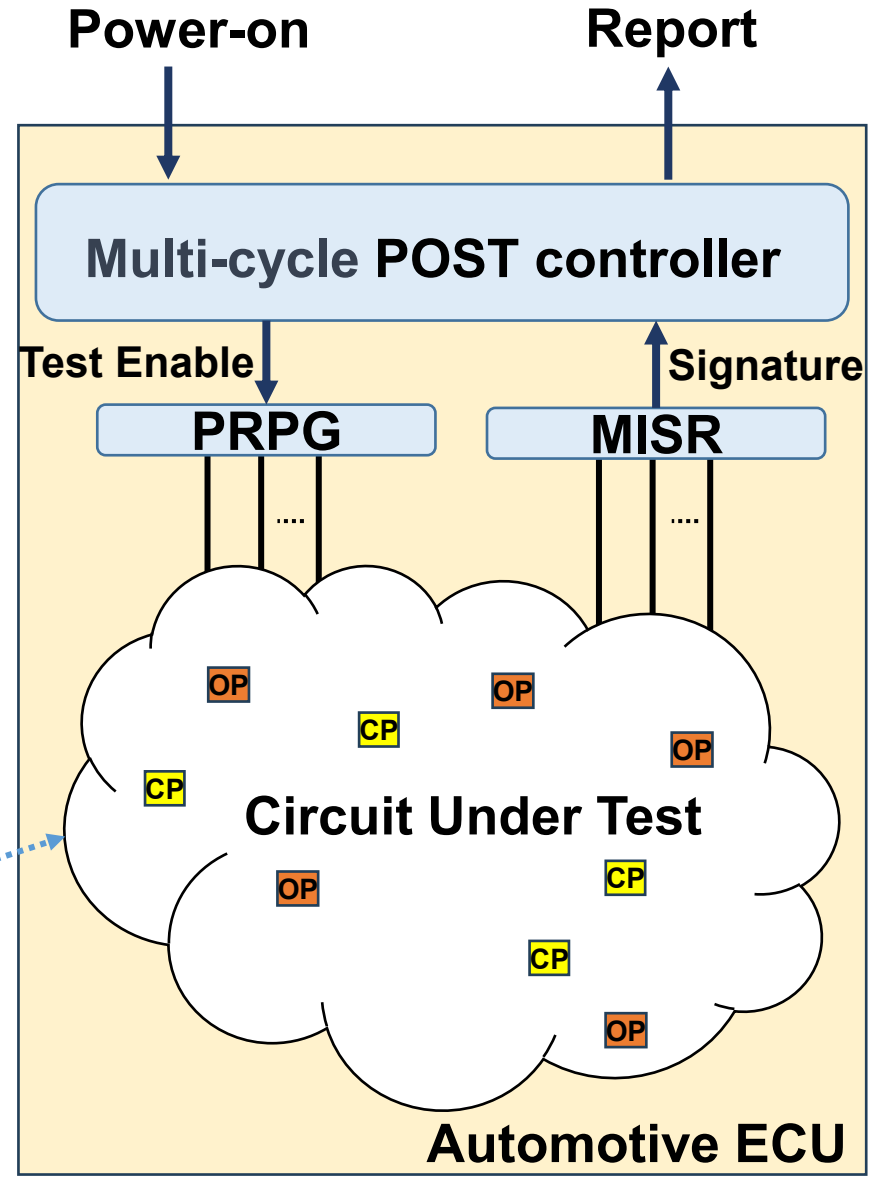
- **Technique 1: Multi-Cycle Power-on Self-Test**
  - **Multi-cycle Test**



- **Technique 2: Test Point Insertion Technique**
  - **Observation Point (OP) & Control Point (CP)**

- **Technique 3: Test Point Selection Optimization**

- **Goal:**
  - **Reduce Test Application Time**
  - **while maintaining high fault coverage**



Multi-cycle Test

- Apply **many captures** to the test that allows **each test pattern** to detect more faults → **Test pattern reduction** (than traditional Scan-Test)

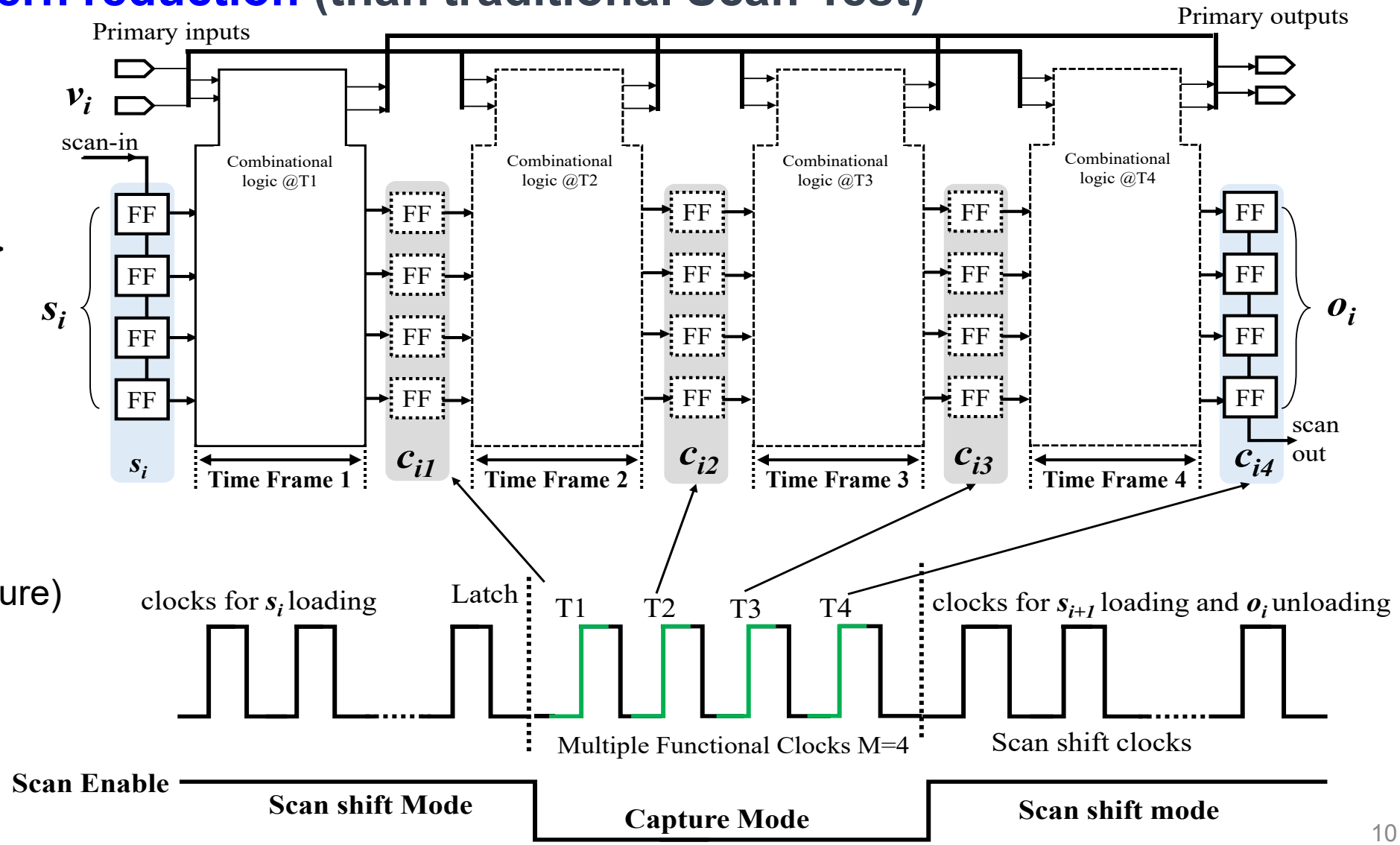
Multi-cycle Test : <s_i, v_i, c_{ij}, o_i>

s_i : scan-in pattern

v_i : primary input vector

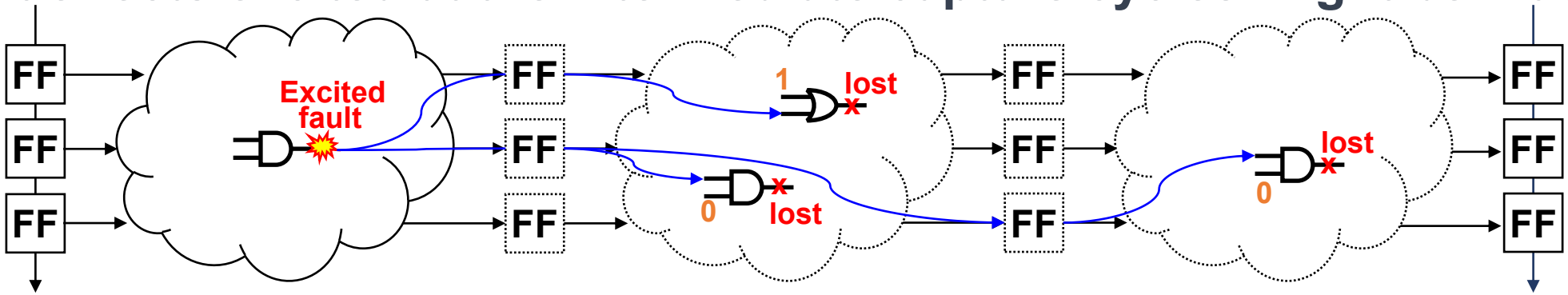
c_{ij} : capture patterns at the jth

o_i : scan-out pattern (last capture)



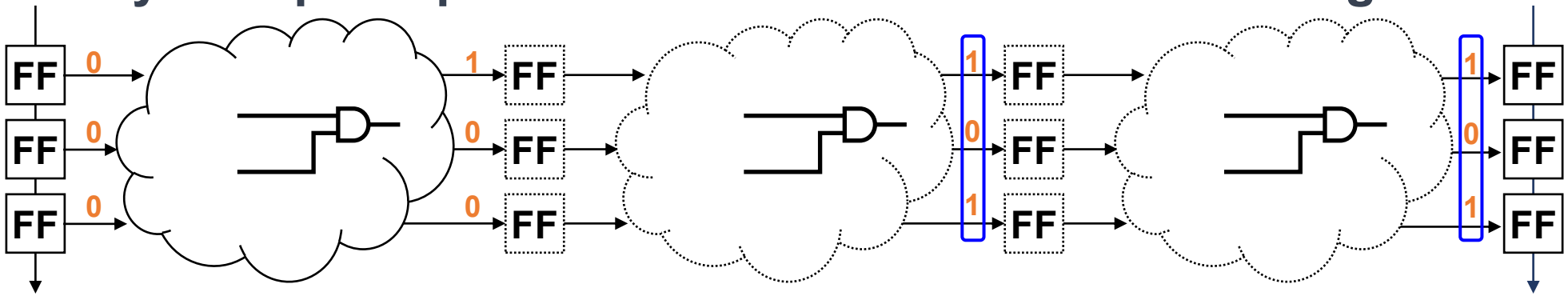
Problems of Multi-cycle Test

- Fault Masking Problem
  - ➢ fault effects excited at the intermediate capture cycles might be masked



Lower observability of signal lines at early capture cycles → Difficult to propagate faults

- Fault Detection Degradation (FDD) Problem
  - ➢ capability of capture patterns to detect additional faults degrades with cycles



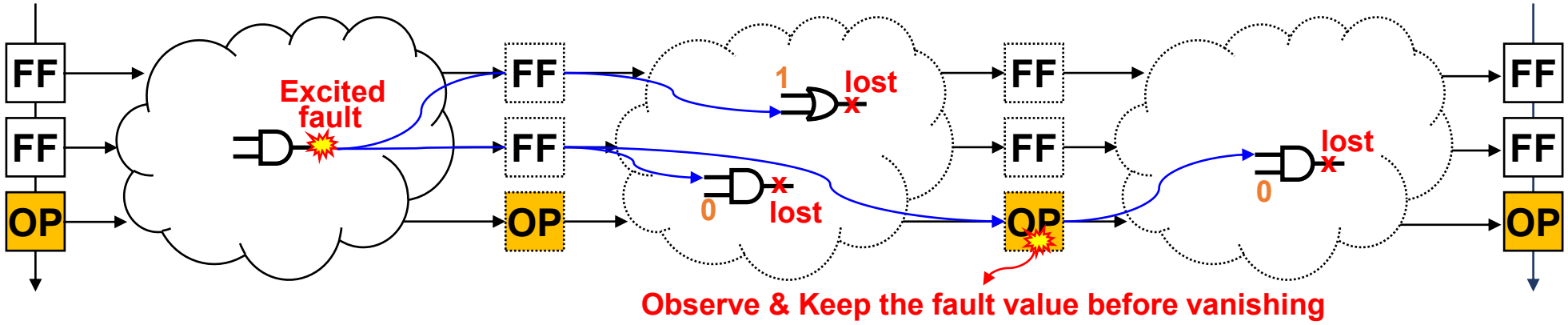
Randomness loss of capture patterns, as capture cycles → Difficult to excite faults

Outline

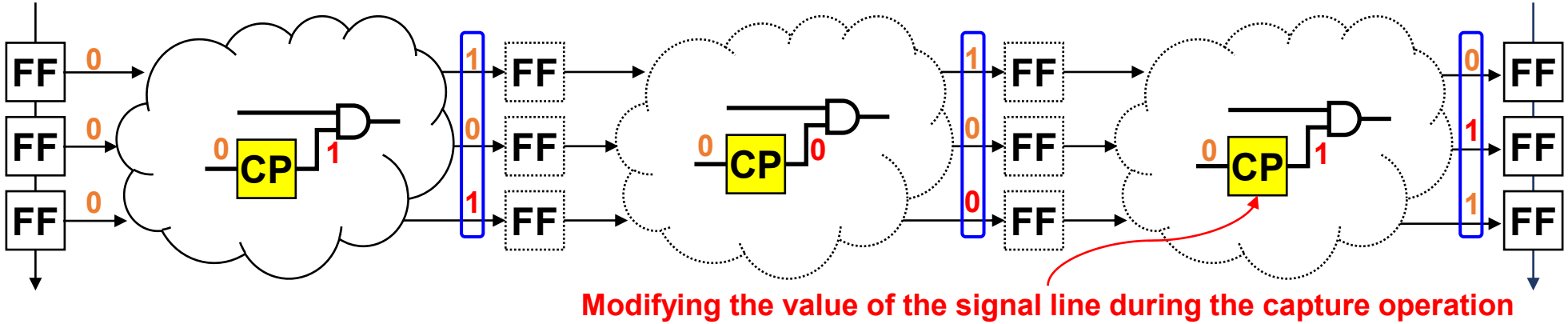
- Background
- Objectives
- **Reliability Enhancement for Automotive ECU**
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Solving Multi-Cycle Test Problems ~ Test Point Insertion ~

- **Solution for Fault Masking**
  - **OPI: Observation Point (OP) Insertion** *for observability*

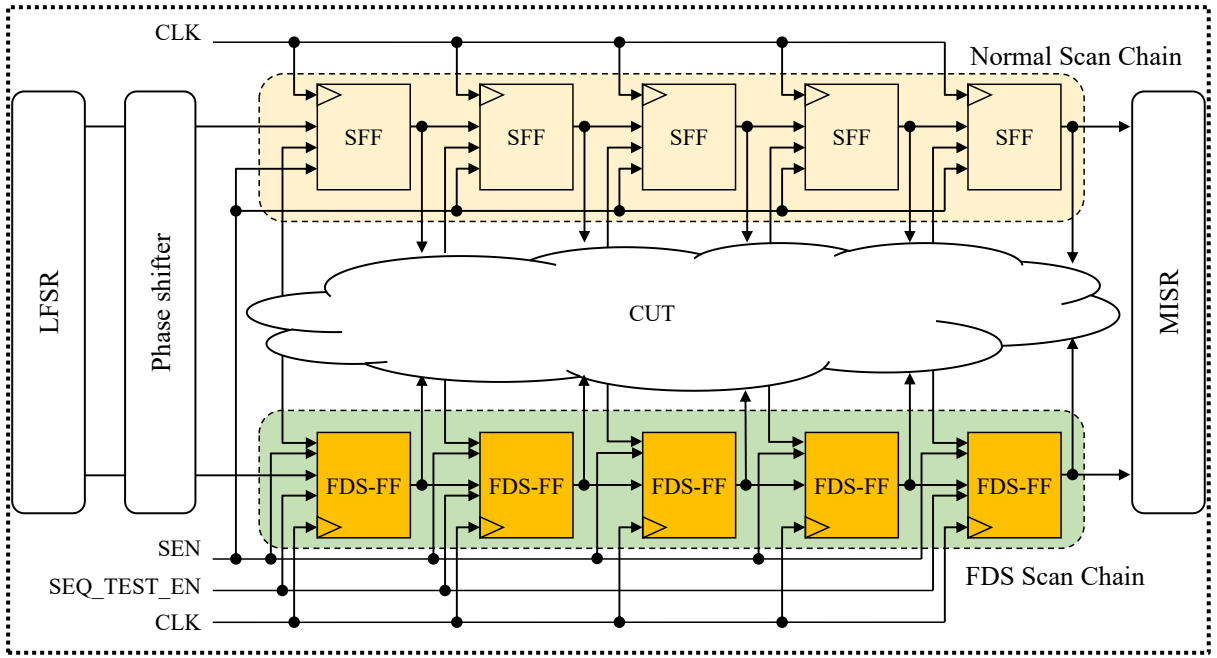


- **Solution for Fault Detection Degradation (FDD)**
  - **CPI: Control Point (CP) Insertion** *for improving controllability*

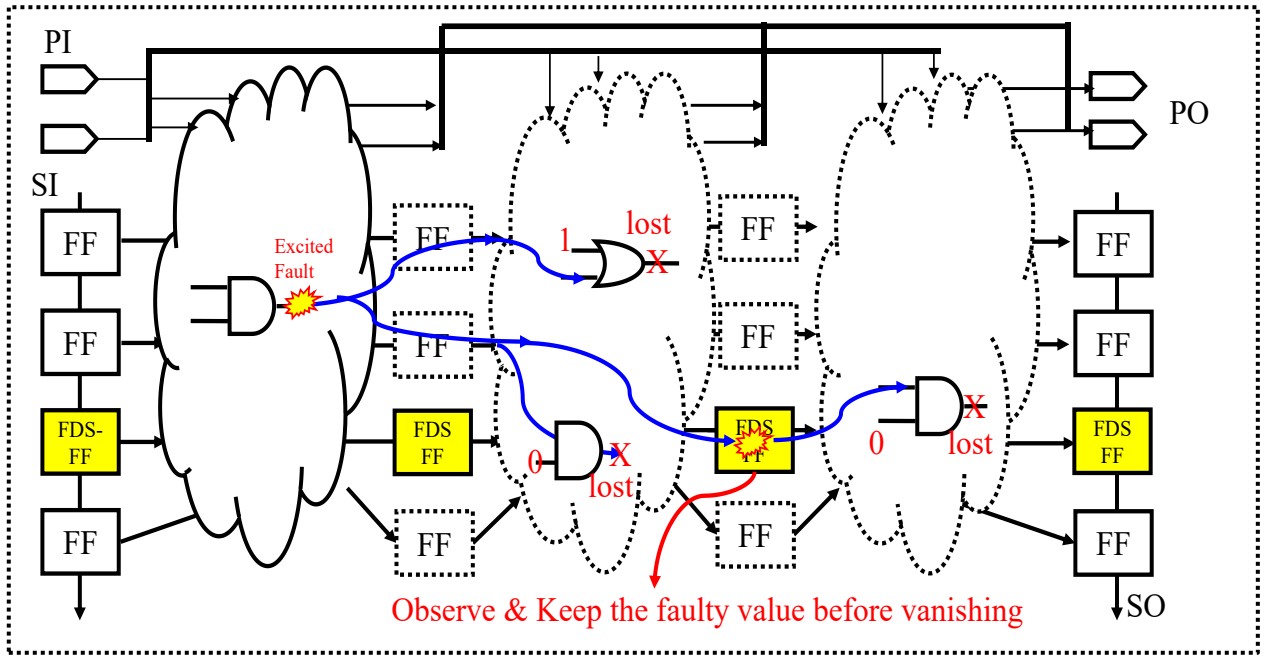


Test Point Insertion ~ Observation Point Insertion (OPI) ~

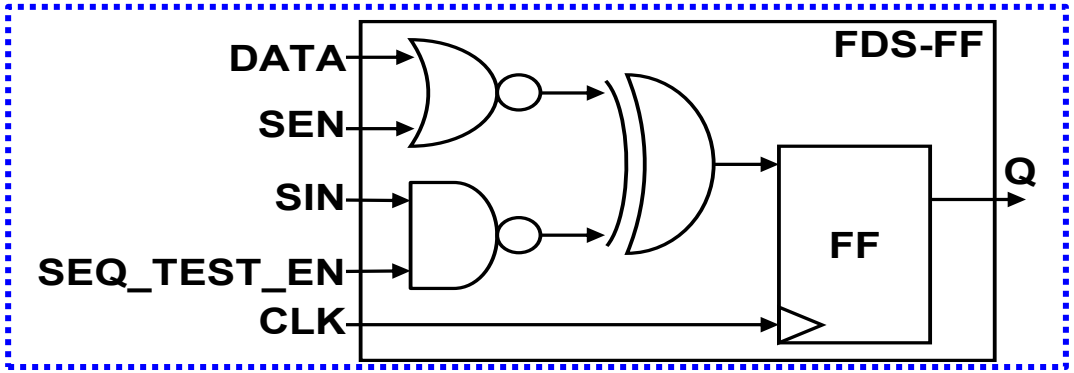
- OP: FDS-FF (fault-detection-strengthened FF)



DFT architecture: LBIST with an independent FDS-FFs scan chain



Replace a scan-FF with FDS-FF to address fault masking



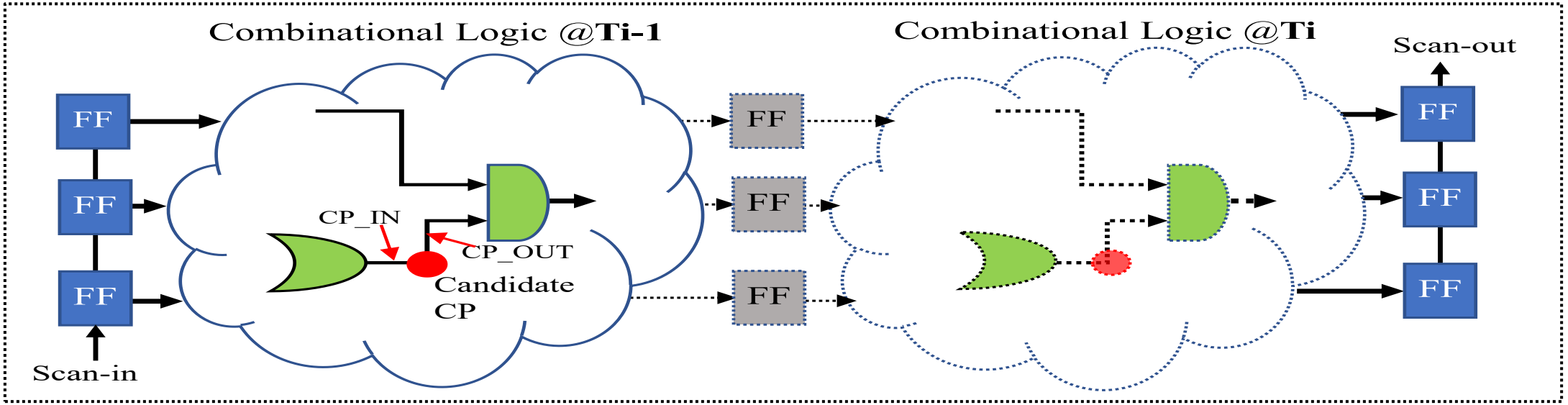
Logic design of FDS-FF

Port	Function	Shift Mode	Capture Mode	User Mode
DATA	Data In	don't care	0/1	0/1
SIN	Scan-in	0/1	0/1	don't care
SEN	Scan Enable	1	0	0
CLK	Clock	010	010	010
SEQ_TEST_EN	Sequential Test Enable	1	1	0
Status		!SIN	DATA^!SIN	DATA

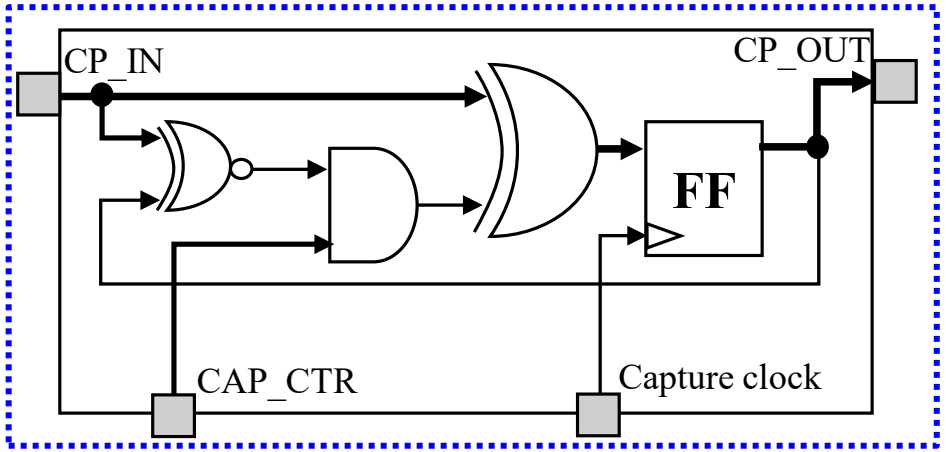
Operation mode of FDS-FF

Test Point Insertion ~ Control Point Insertion (CPI) ~

- CP: Self-Flipping CP



Inserting a Self-Flipping Control Point into the time-expanded circuit to address Fault Detection Degradation



Logic design of Self-Flipping CP

CAP_CTR	CP_OUT @Ti-1	CP_IN @Ti	CP_OUT @Ti
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1

Truth table of self-flipping control logic

TP Selection Procedure for Multi-cycle Test

- The procedure consists of two phases
  - 1: CP insertion under a time-expanded circuit with full FF-observation
  - 2: OP pruning to remove the impotent observation points (FDS-FF)

Phase 1: CP insertion

Read the original CUT Netlist

↓

Time expansion for the CUT Netlist

↓

Full observation point insertion

↓

Evaluation and Select CP

↓

Insert CP to CUT Netlist

↓

Out CP and CUT Netlist with CP insertion

Phase 2: OP Pruning

Read CUT Netlist with CP insertion

↓

OP list (all FFs)

↓

Analysis Netlist Structure

↓

Rank FFs for the OP list

↓

Evaluation and Remove OP

↓

Out OP and CUT Netlist with OP Pruning

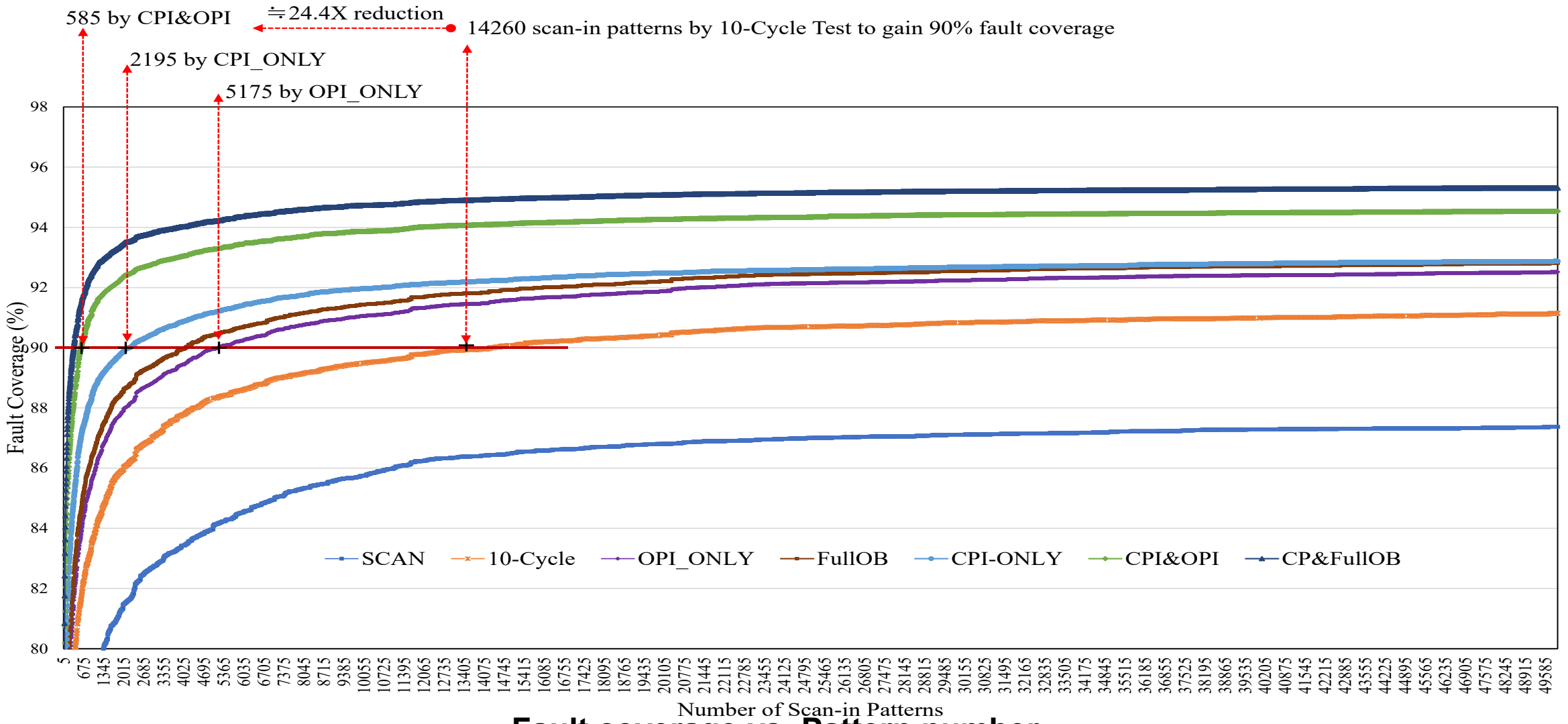
Experimental Setting

- **Procedure(TP insertion & selection):**
  - C Program Language
- **Experimental data (circuit netlist):**
  - ISCAS89, ITC99 benchmark
- **Fault model:**
  - Stuck-at fault
- **Maximum number of TPs:**
  - # CP<1% of gates & <5% of FFs
  - # OP<20% of FFs
- **Computer:**
  - OS: Ubuntu18.4
  - CPU: Intel Xeon W-2245, 64GB memory

	Circuit	# gate	# FF	# Fault
ISCAS89	s9234	5597	228	6927
	s13207	7951	669	9815
	s15850	9772	597	11725
	s38417	22179	1636	31180
	s38584	19253	1452	36303
ITC99	b11	437	31	1322
	b12	904	121	2797
	b14	4444	245	12811
	b15	8338	449	23528
	b17	22645	1415	65464
	b20	8875	490	25338

Experimental Results ~ Efficiency of the CPI and the OPI ~

- The number of scan-in patterns for achieving 90% fault coverage is drastically reduced to 585 (24.4X reduction compared to the multi-cycle test)



Fault coverage vs. Pattern number

Outline

- Background
- Objectives
- **Reliability Enhancement for Automotive ECU**
  - **Multi-Cycle Power-on Self-Test (POST)**
  - **Test Point Insertion Technique for Multi-Cycle POST**
  - **Conclusions**
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Conclusions ~ Enhancing the Reliability of Automotive ECU ~

Functional Safety for Automotive ECU:

- Need for In-Field Testing: Power-On Self-Test (POST)
- Challenge of Achieving Efficient POST

  Limited test application time (<50ms) & High fault coverage (>90%)

Approach:

- **Technique 1**: Multi-Cycle Power-on Self-Test
- **Technique 2**: Test Point Insertion Technique
- **Technique 3**: Optimization for Test Point Selection
- Goal:
  - Reduce Test Application Time
  - while maintaining high Fault Detection Quality

Results:

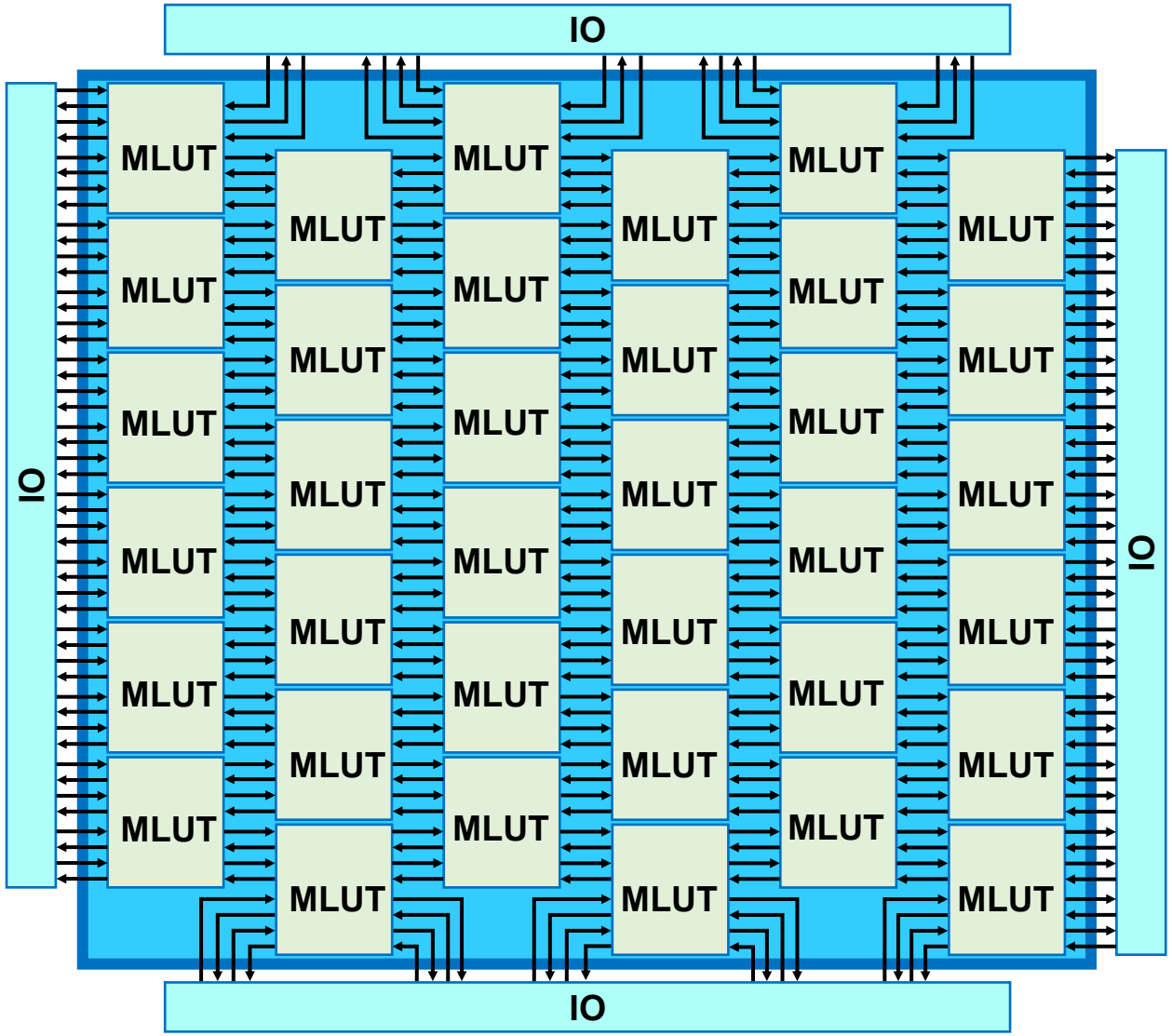
- 24.4X reduction in scan-in patterns for achieving 90% fault coverage

Outline

- Background
- Objectives
- Reliability Enhancement for Automotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- **Reliability Enhancement for MPLD**
  - **Interconnect Defect Testing**
  - **Aging Monitoring Techniques**
  - **Conclusions**
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

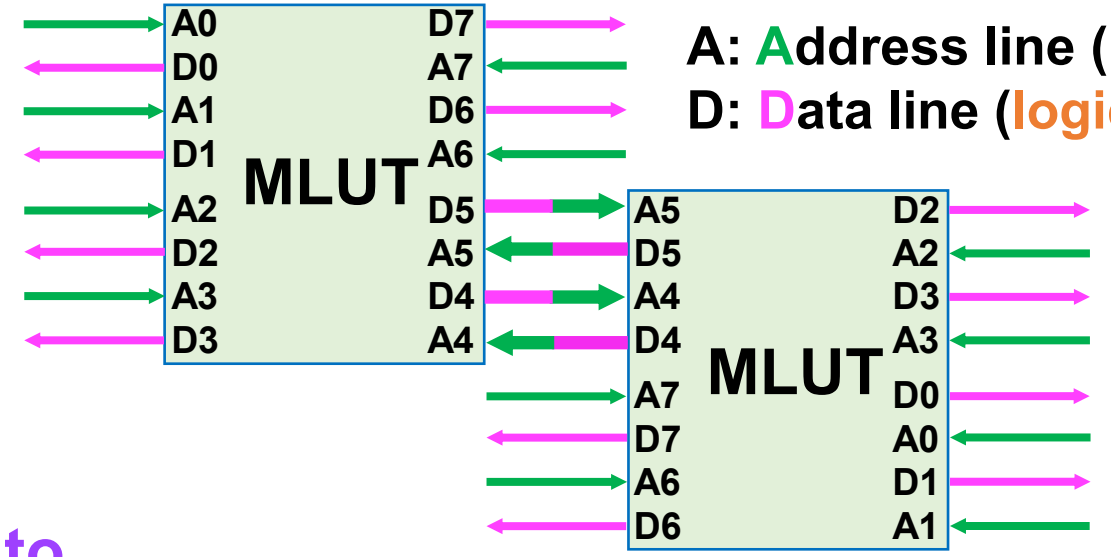
What's MPLD ~ Architecture~

- A new type reconfigurable device
- Memory-based Programmable Logic Device (MPLD)
- constructed only by MLUT (Multiple Look-Up-Table) array in a special interconnect structure.



What's MPLD ~ Architecture ~ --- AD-pair Interconnect Structure

- Address lines and Data lines alternately connect with others



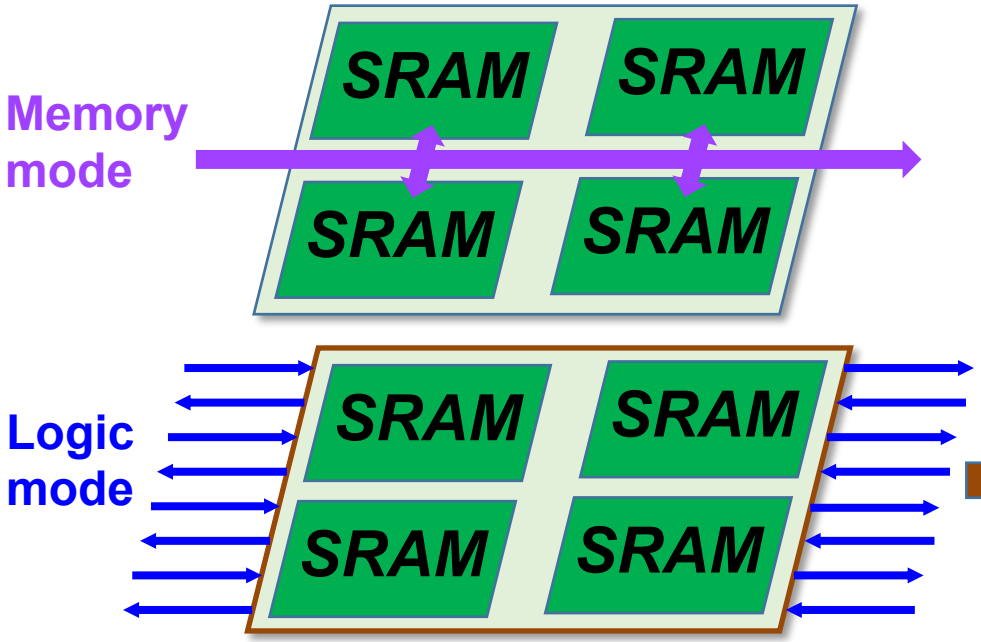
A: Address line (logic)
D: Data line (logic)

- Logic data output of a MLUT connects to address input of its neighbor MLUTs

Interconnect Structure
(AD-pair Interconnect)

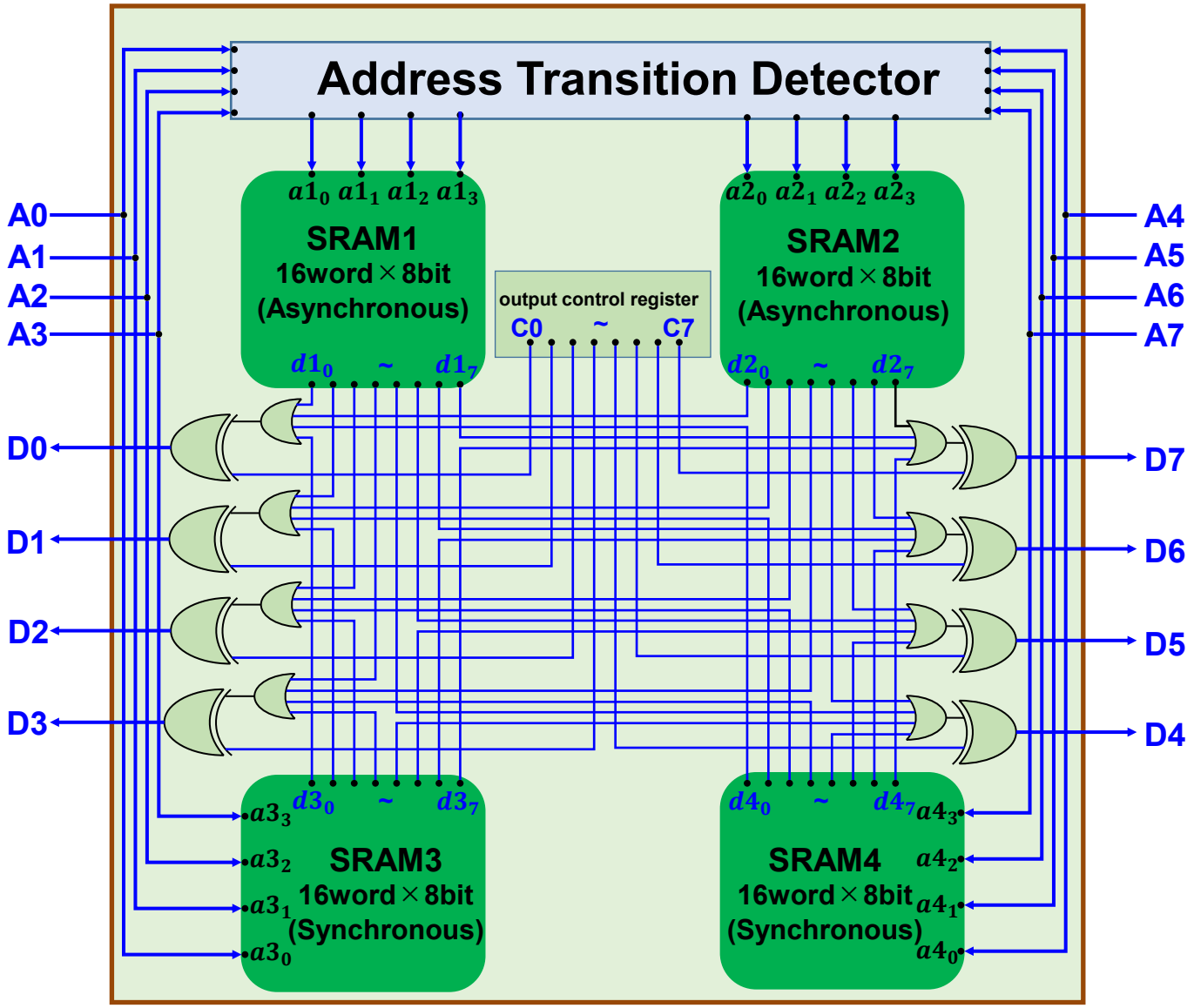
What's MPLD ~ Architecture ~ --- MLUT structure

- basic reconfigurable elements
- multiple SRAM blocks
- memory mode or logic mode
- each SRAM works as LUTs



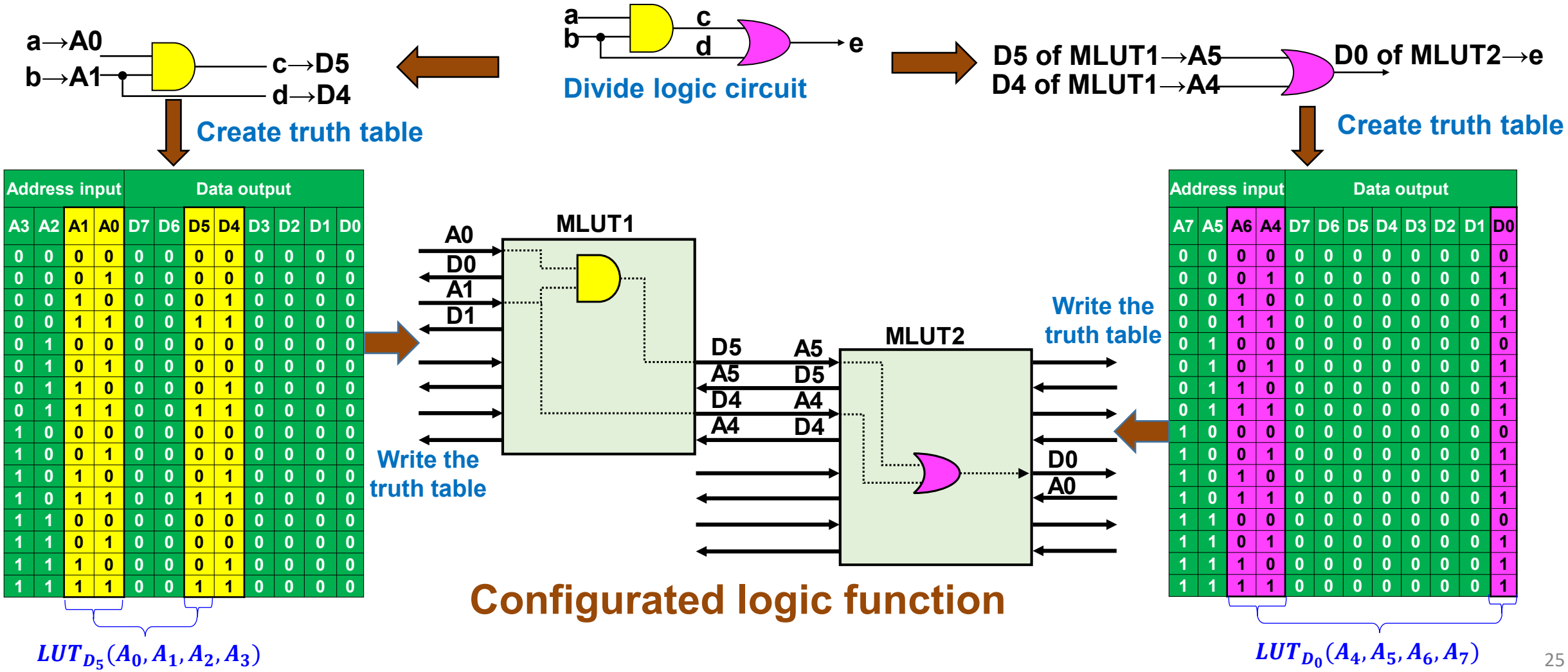
$$D_n = C_n \oplus (d1_n \cup d2_n \cup d3_n \cup d4_n)$$

$$di_n = LUT_{di_n}(ai_0, ai_1, ai_2, ai_3)$$



What's MPLD ~ Architecture ~ --- AD-pair Interconnect Structure

- Configure the logic function by writing the truth table of the logic circuit (including wiring logic) into the SRAM of MLUT

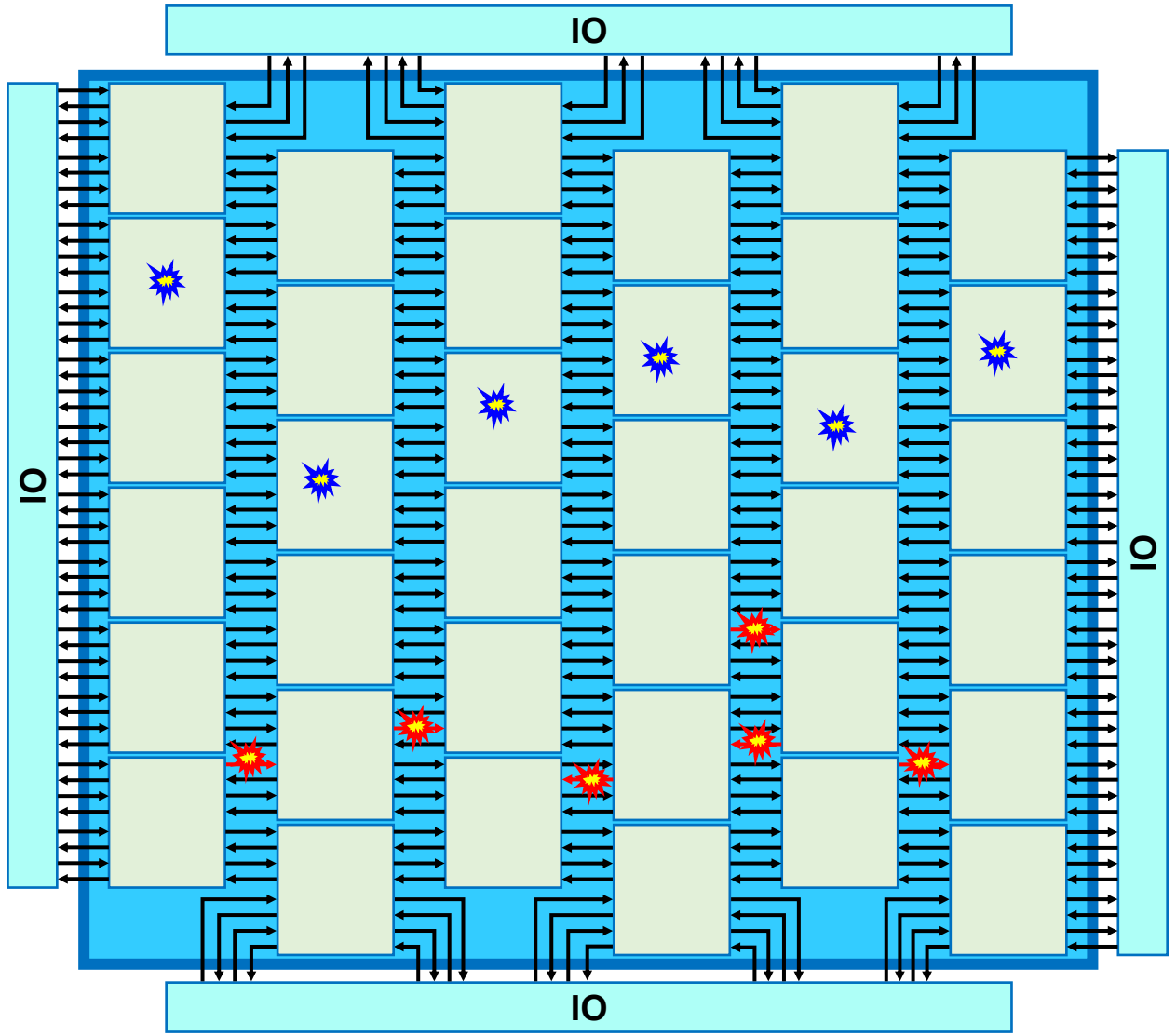


Reliability Issue in MPLD ~ Manufacturing ~

Factors:

- Manufacturing Phase
  - Defect in MLUT (SRAM)
    - ✓ Conventional Memory testing
  - Defect between MLUTs
    - ➢ Interconnect defect on Address and Data lines (short, bridge, open, etc.)

yield loss and reliability degradation

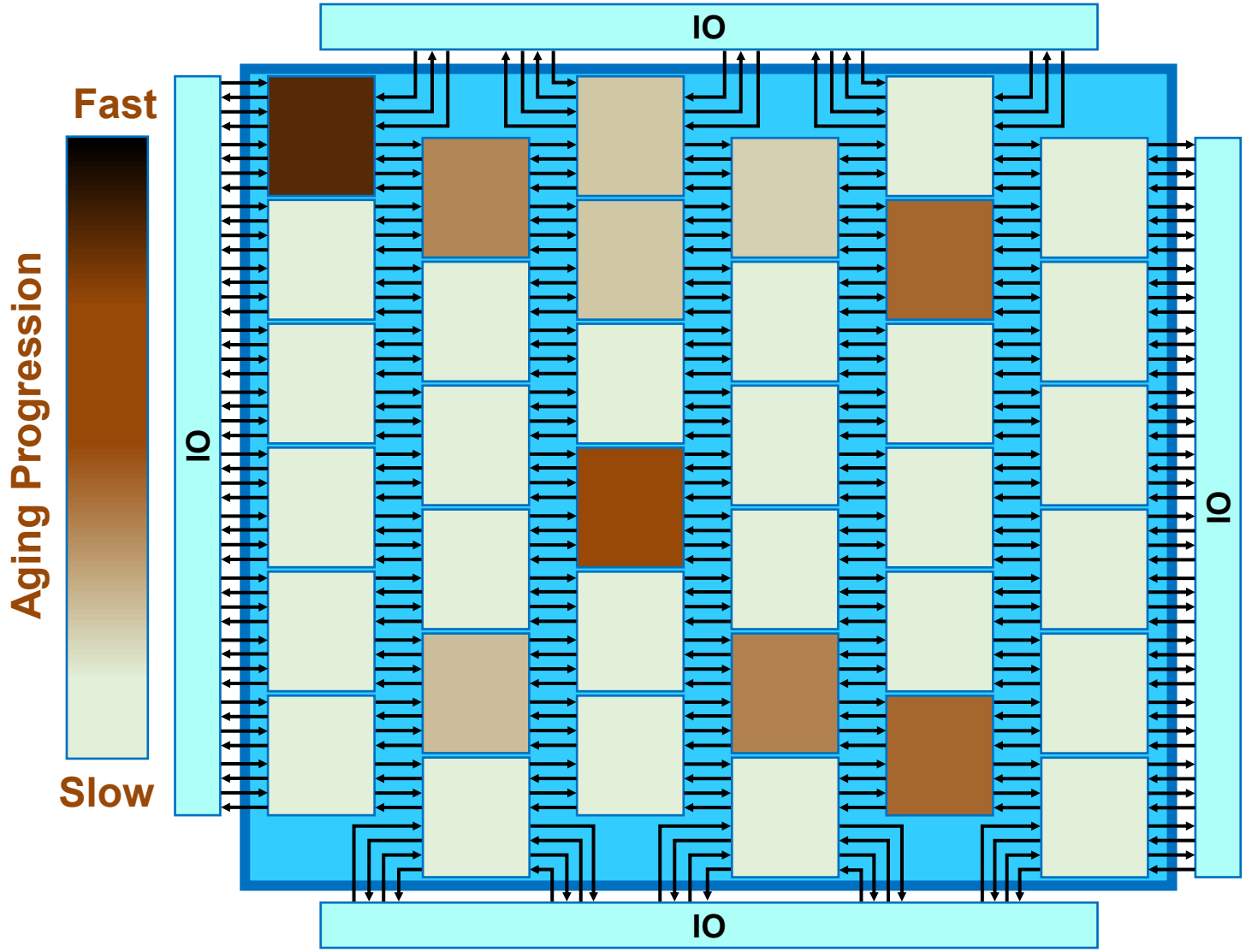


Reliability Issue in MPLD ~ In-Filed Aging ~

Factors:

- Application phase (in the field)
  - Aging in MLUTs
    - ➢ HCI, BTI, etc.
    - ➢ Aging-induced delay
  - Different aging progress
    - ➢ system failure
    - ➢ logic circuit performance

    (e.g.: sudden system down/reset)



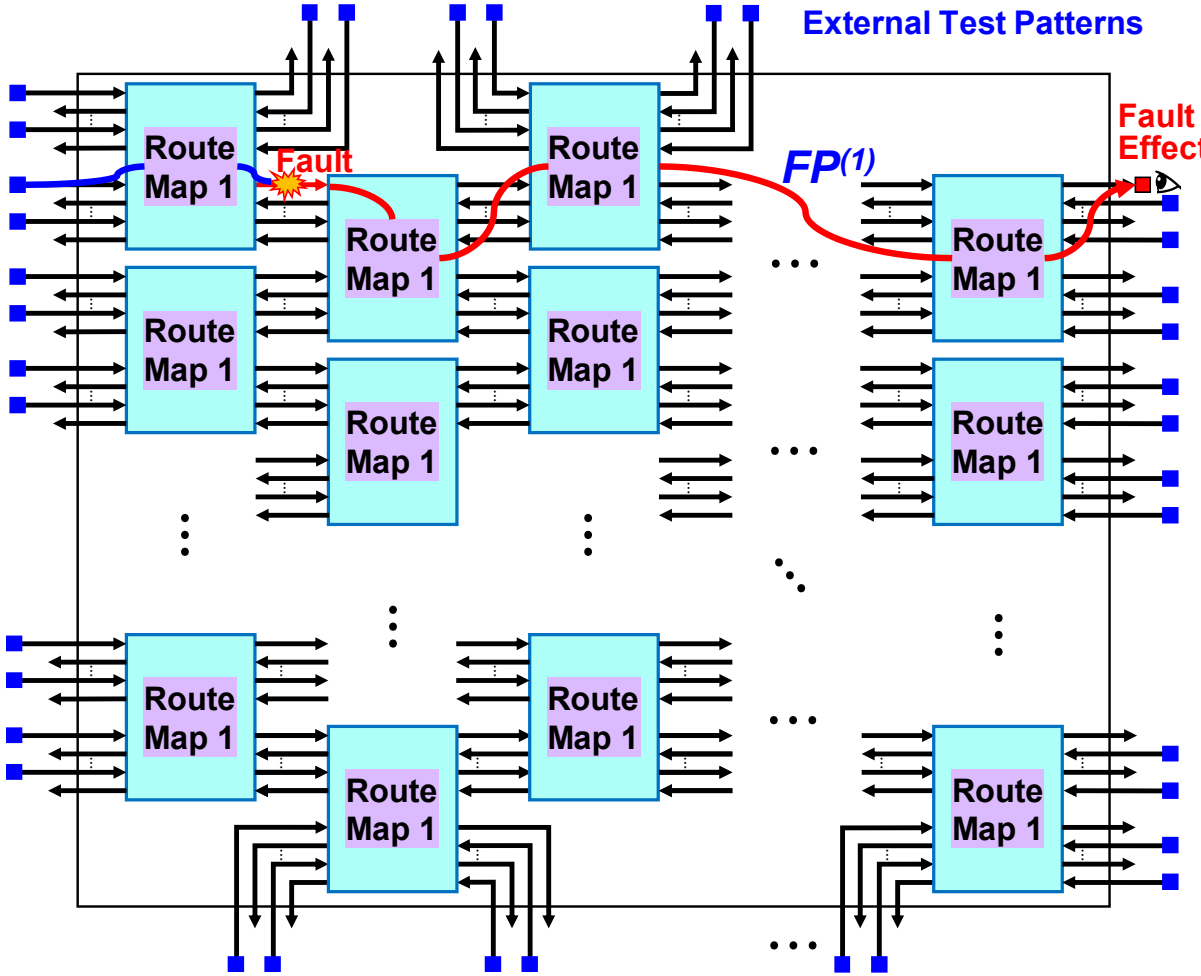
Outline

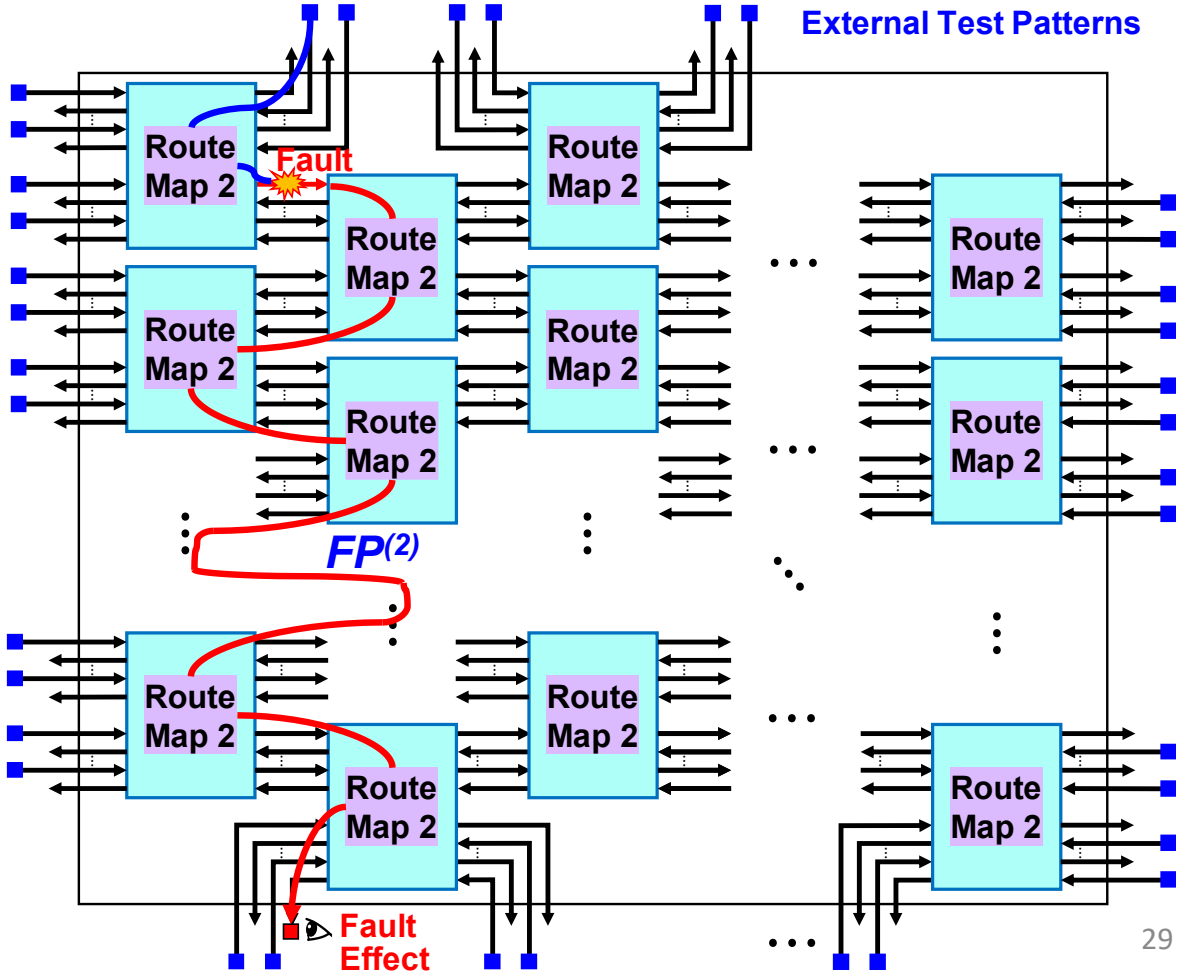
- Background
- Objectives
- Reliability Enhancement for Automotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- **Reliability Enhancement for MPLD**
  - **Interconnect Defect Testing**
  - **Aging Monitoring Techniques**
  - **Conclusions**
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Manufacturing Defect Testing ~ Basic Idea ~

- Create Route Maps on the MLUTs array for fault propagation
- Apply external test patterns to external input ports to excite the target faults
- Detect fault by Fault Effects at external output ports
- Locate faults by the intersection of Fault Paths under different route maps

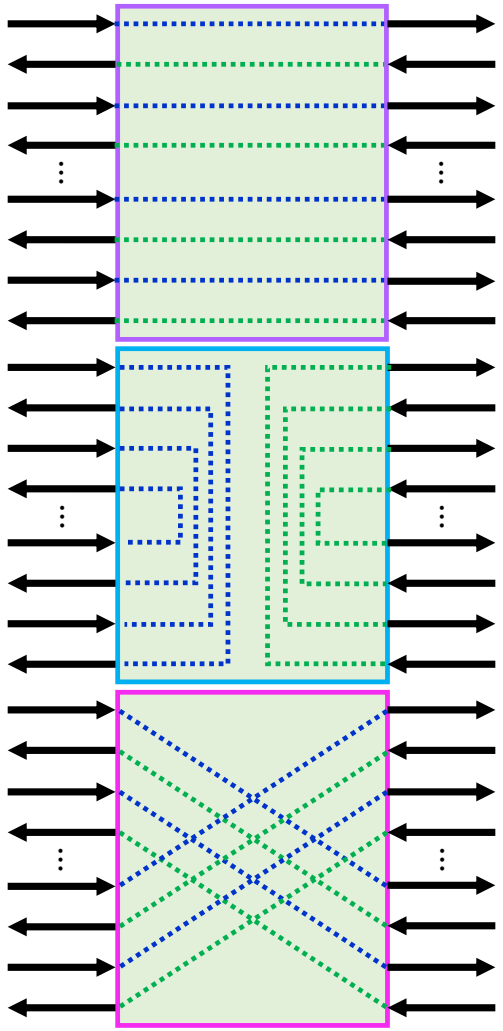
$$F_{loc}=FP^{(1)}\cap FP^{(2)}$$





Manufacturing Defect Testing ~ Test Cube ~

- **Route Map (rm)** is created by **Test Cube (TC)** stored into SRAM of MLUTs
- **truth table1** route **low-order address**; **truth table2** route **high-order address**



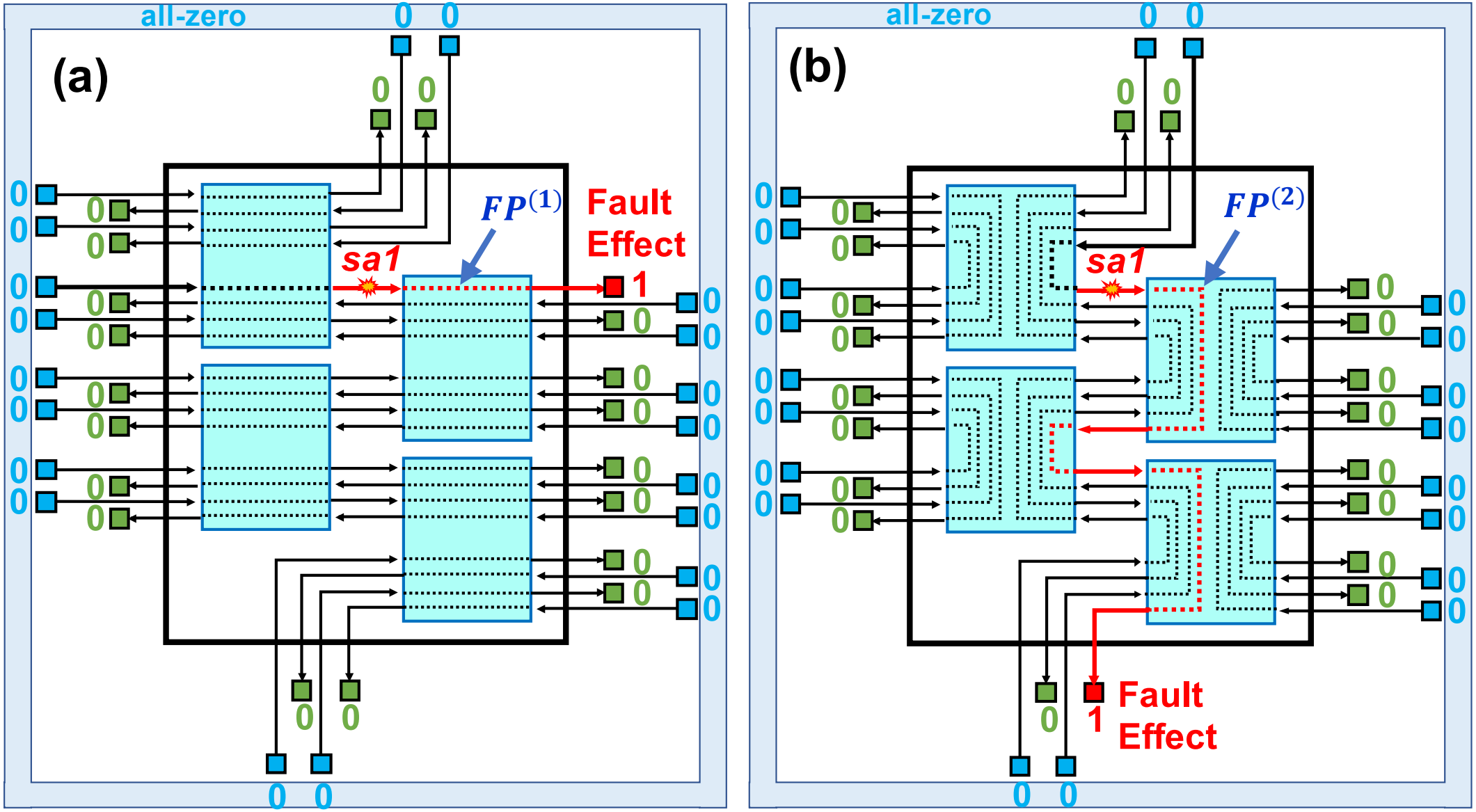
Route Maps	Test Cubes		
rm_1: horizontal route map	$TC^{(1)}$	truth table1	$D_{m-1:m/2} = A_{0:m/2-1}$
			$D_{m/2-1:0} = all\text{-}0$
		truth table2	$D_{m-1:m/2} = all\text{-}0$
			$D_{m/2-1:0} = A_{m/2:m-1}$
rm_2: vertical route map	$TC^{(2)}$	truth table1	$D_{m-1:m/2} = all\text{-}0$
			$D_{m/2-1:0} = A_{0:m/2-1}$
		truth table2	$D_{m-1:m/2} = A_{m/2:m-1}$
			$D_{m/2-1:0} = all\text{-}0$
rm_3: diagonal route map	$TC^{(3)}$	truth table1	$D_{m-1:m/2} = A_{m/4:m/2-1}:A_{0:m/4-1}$
			$D_{m/2-1:0} = all\text{-}0$
		truth table2	$D_{m-1:m/2} = all\text{-}0$
			$D_{m/2-1:0} = A_{3m/4:m-1}:A_{m/2:3m/4-1}$

Manufacturing Defect Testing ~ External Test Pattern~

- **External Test Patterns for exciting the *stuck-at* and *bridge* interconnect faults by applying *all-zero/one, walking-zero/one vectors*.**

Fault Types	External Test Patterns
stuck-at-1	*all-zero vector: 0…0*
stuck-at-0	*all-one vector: 1…1*
AND-bridge	*walking-zero vector: 1…1$\vec{0}$1…1*
OR-bridge	*walking-one vector: 0…0$\vec{1}$0…0*

Manufacturing Defect Testing ~ example ~ --- testing stuck-at-1 fault



$F_{loc}=FP^{(1)} \cap FP^{(2)}$

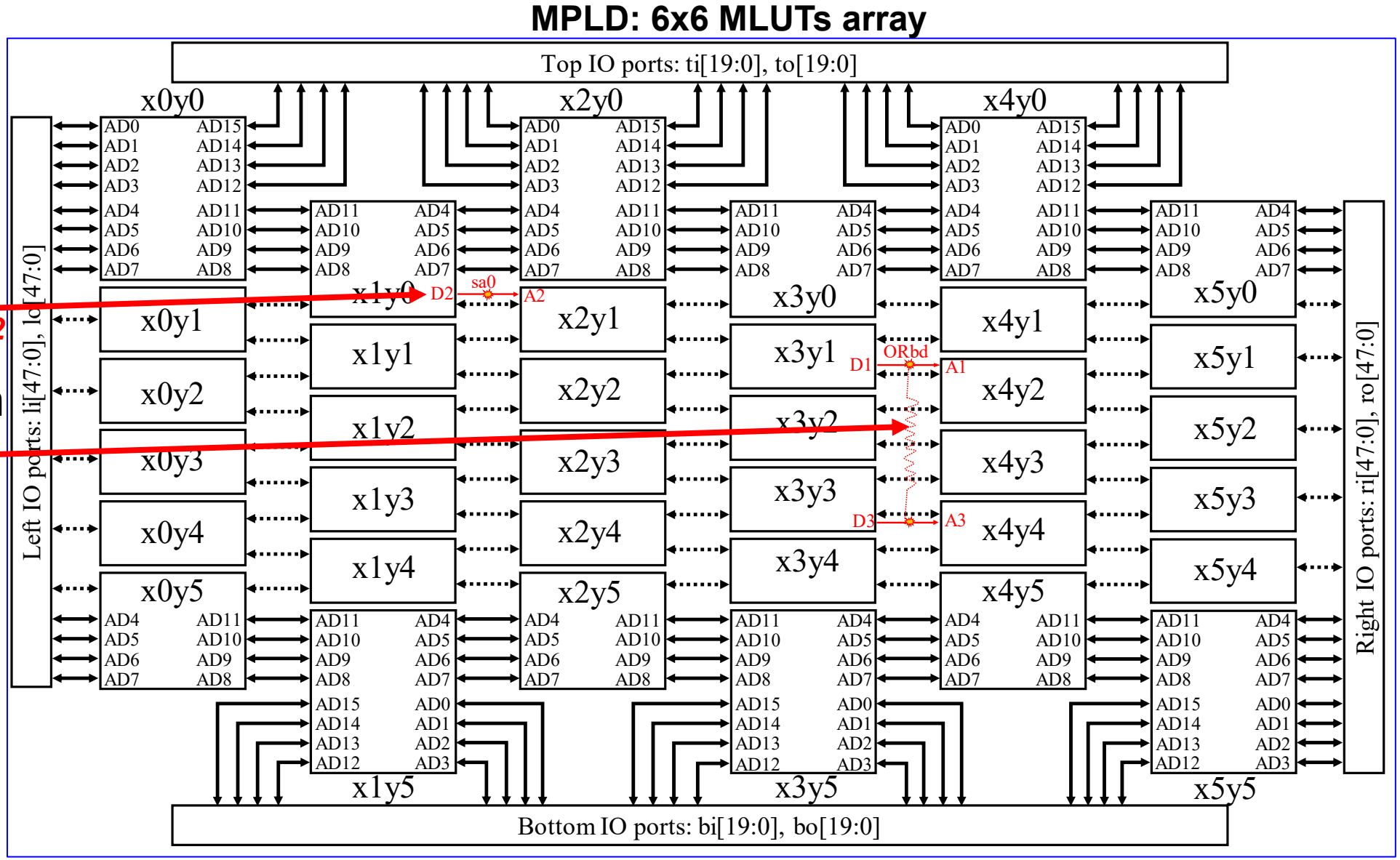
Manufacturing Defect Testing ~ simulation setting ~

- **MPLD Design**:
  Verilog HDL

- **Fault injection**:
  - Stuck-at-0 at $x_2y_1A_2$
  - OR-bridge between x4y2A1 , x4y4A3

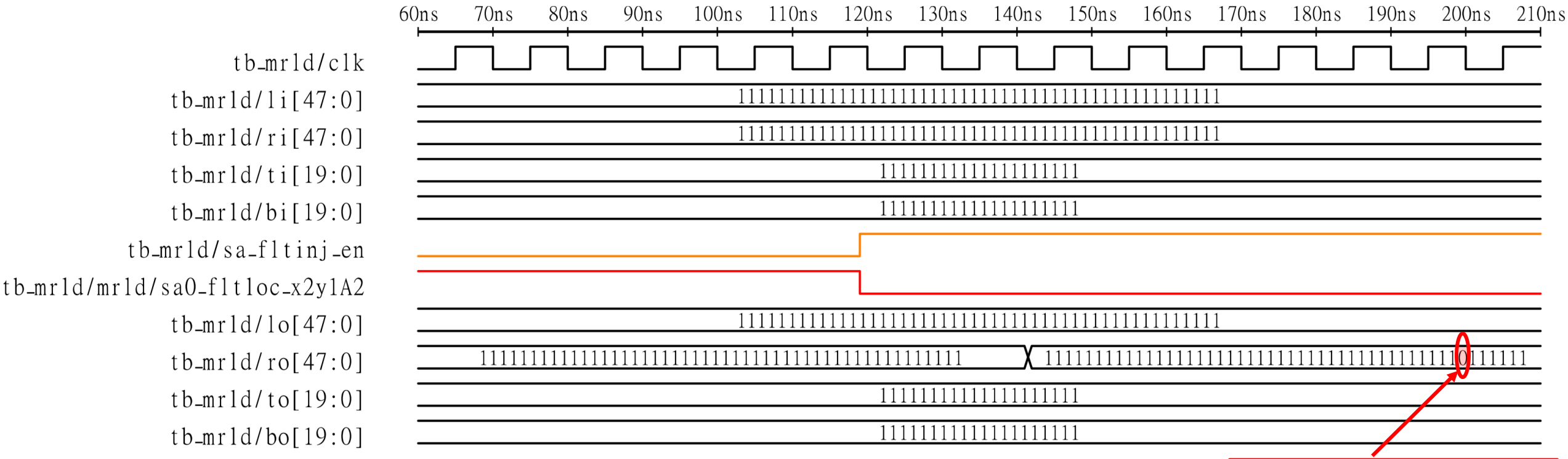
- **Simulation Tool**:
  ModelSim

- **Computer**:
  - OS: Windows 10 Home
  - CPU: Intel Core i9-10900, 32GB memory



Manufacturing Defect Testing ~ simulation ~ --- stuck-at-0 fault

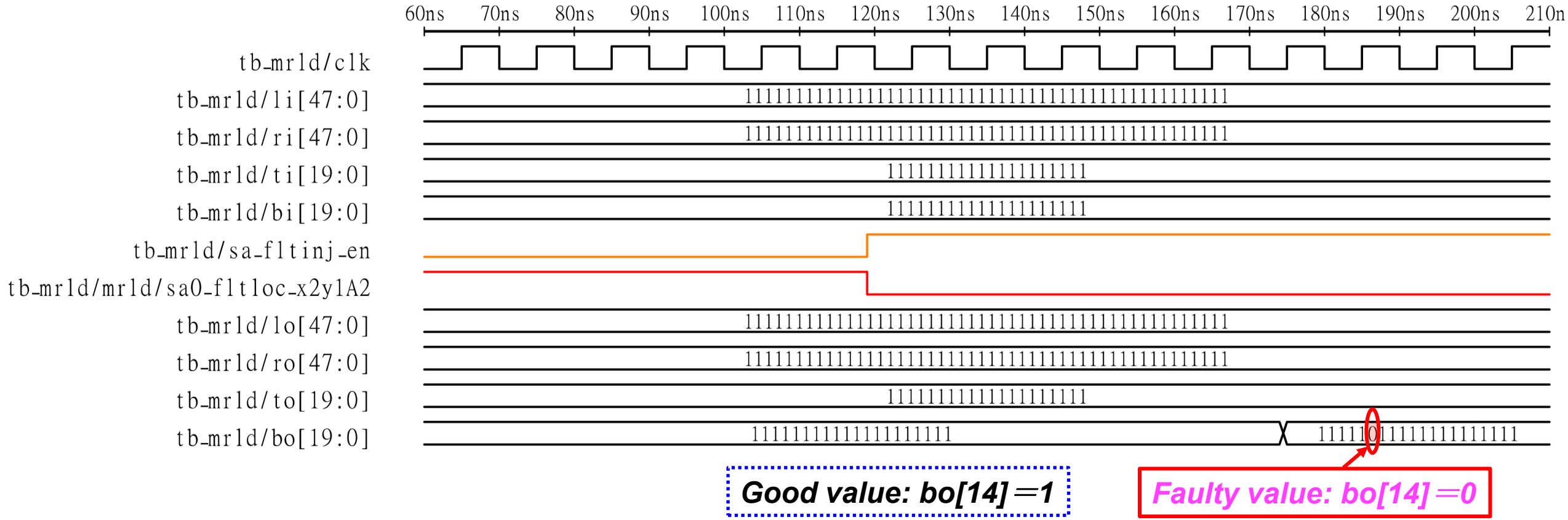
Testing under Horizontal Route Map



$FP^{(1)}=\{li[10]\rightarrow x_1y_0A_{13}\rightarrow \mathbf{x_2y_1A_2}\rightarrow x_3y_0A_{13}\rightarrow x_4y_1A_2\rightarrow x_5y_0A_{13}\rightarrow ro[6]\}$,

Manufacturing Defect Testing ~ simulation ~ --- stuck-at-0 fault

Testing under Vertical Route Map

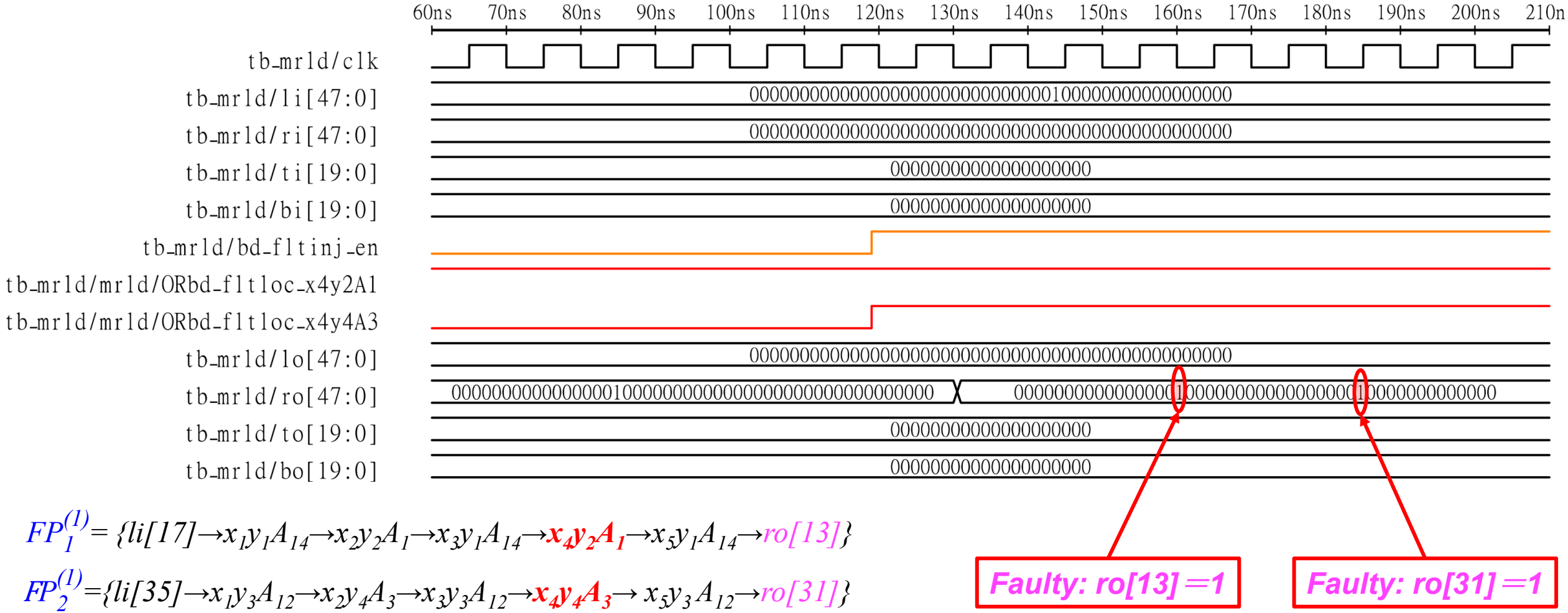


$FP^{(2)}=\{ti[14]\rightarrow x_1y_0A_5\rightarrow \mathbf{x_2y_1A_2}\rightarrow x_1y_1A_5\rightarrow x_2y_2A_2\rightarrow x_1y_2A_5\rightarrow x_2y_3A_2\rightarrow x_1y_3A_5\rightarrow x_2y_4A_2\rightarrow x_1y_4A_5\rightarrow x_2y_5A_2\rightarrow x_1y_5A_5\rightarrow bo[14]\}$

$$F_{loc}=\bigcap_{i=1}^{2} FP^{(i)}=FP^{(1)}\cap FP^{(2)}=\mathbf{x_2y_1A_2}$$

Manufacturing Defect Testing ~ simulation ~ --- OR-bridge fault

Testing under Horizontal Route Map



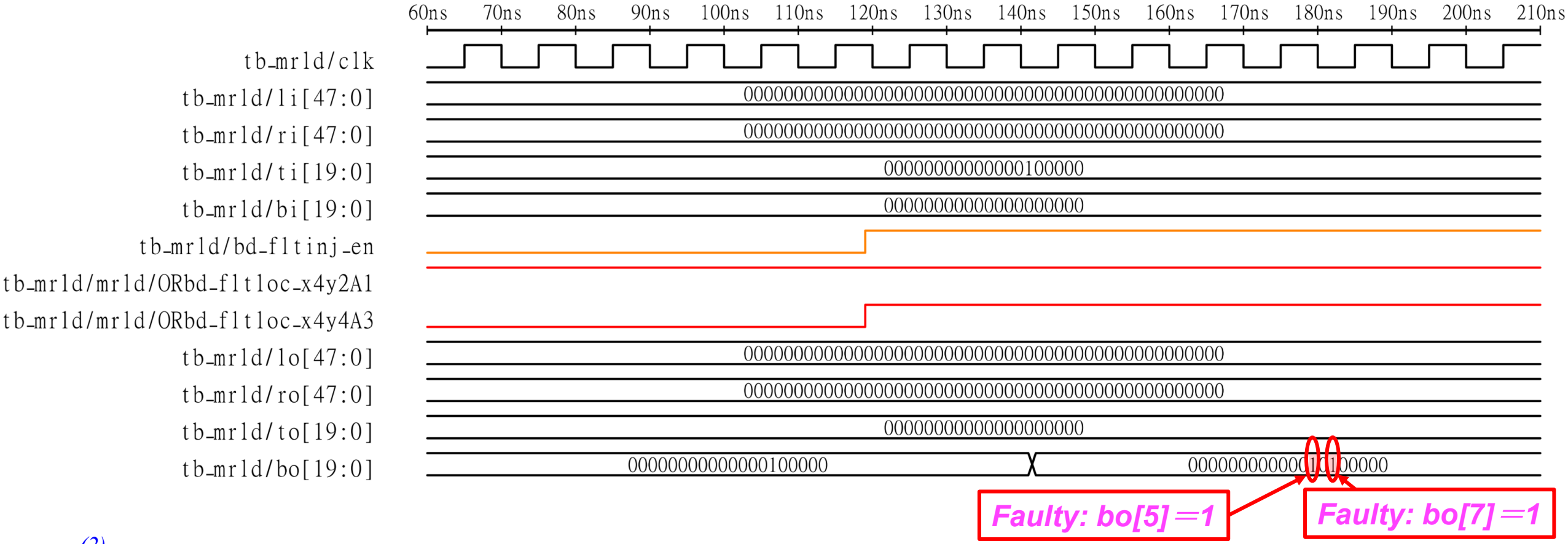
$FP_1^{(1)} = \{li[17] \rightarrow x_1y_1A_{14} \rightarrow x_2y_2A_1 \rightarrow x_3y_1A_{14} \rightarrow \mathbf{x_4y_2A_1} \rightarrow x_5y_1A_{14} \rightarrow ro[13]\}$

$FP_2^{(1)} = \{li[35] \rightarrow x_1y_3A_{12} \rightarrow x_2y_4A_3 \rightarrow x_3y_3A_{12} \rightarrow \mathbf{x_4y_4A_3} \rightarrow x_5y_3A_{12} \rightarrow ro[31]\}$

$FP^{(1)} = \cup_{k=1}^{2} FP_k^{(1)} = FP_1^{(1)} \cup FP_2^{(1)}$

Manufacturing Defect Testing ~ simulation ~ --- OR-bridge fault

Testing under Vertical Route Map



$FP_1^{(2)}$ = {ti[5] → $x_3y_0A_6$ → $x_4y_1A_1$ → $x_3y_1A_6$ → $\mathbf{x_4y_2A_1}$ → $x_3y_2A_6$ → $x_4y_3A_1$ → $x_3y_3A_6$ → $x_4y_4A_1$ → $x_3y_4A_6$ → $x_4y_5A_1$ → $x_3y_5A_6$ → bo[5]}

$FP_2^{(2)}$ = {ti[7] → $x_3y_0A_4$ → $x_4y_1A_3$ → $x_3y_1A_4$ → $x_4y_2A_3$ → $x_3y_2A_4$ → $x_4y_3A_3$ → $x_3y_3A_4$ → $\mathbf{x_4y_4A_3}$ → $x_3y_4A_4$ → $x_4y_5A_3$ → $x_3y_5A_4$ → bo[7]}

$FP^{(2)} = \cup_{k=1}^{2} FP_k^{(2)} = FP_1^{(2)} \cup FP_2^{(2)}$

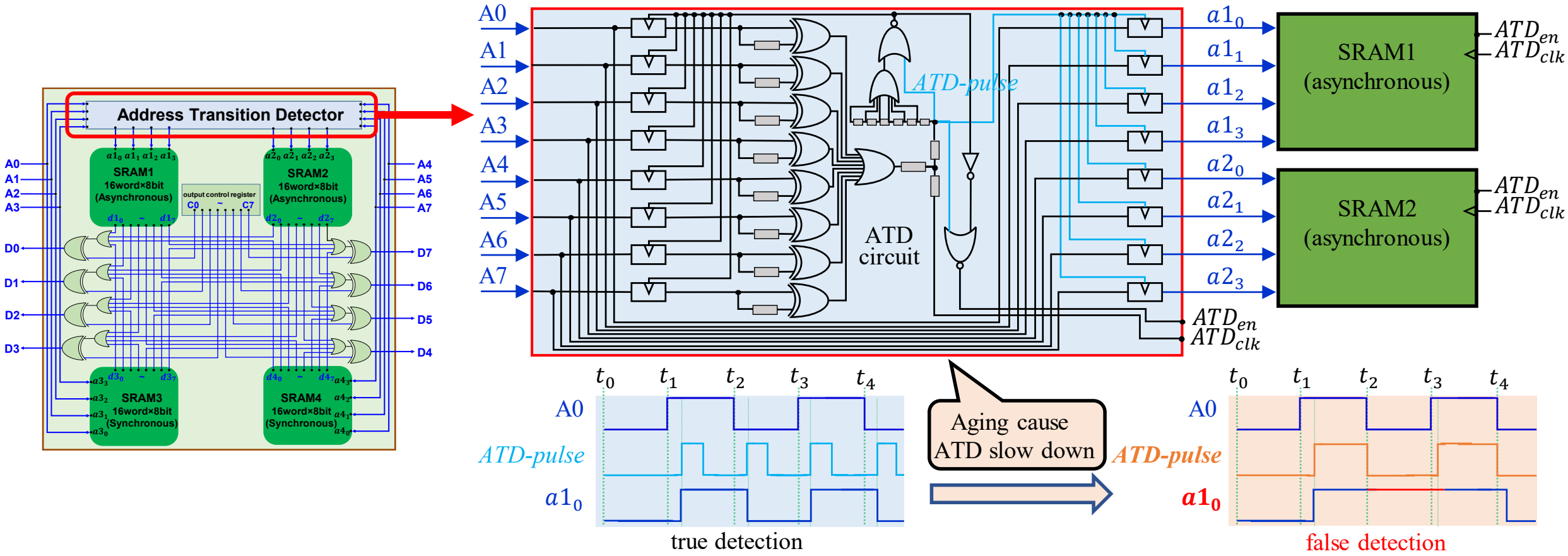
$F_{loc} = \cap_{i=1}^{2} FP^{(i)} = FP^{(1)} \cap FP^{(2)} = \{\mathbf{x_4y_2A_1}, \mathbf{x_4y_4A_3}\}$

Outline

- Background
- Objectives
- Reliability Enhancement forAutomotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- **Reliability Enhancement forMPLD**
  - **Interconnect Defect Testing**
  - **Aging Monitoring Techniques**
  - **Conclusions**
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

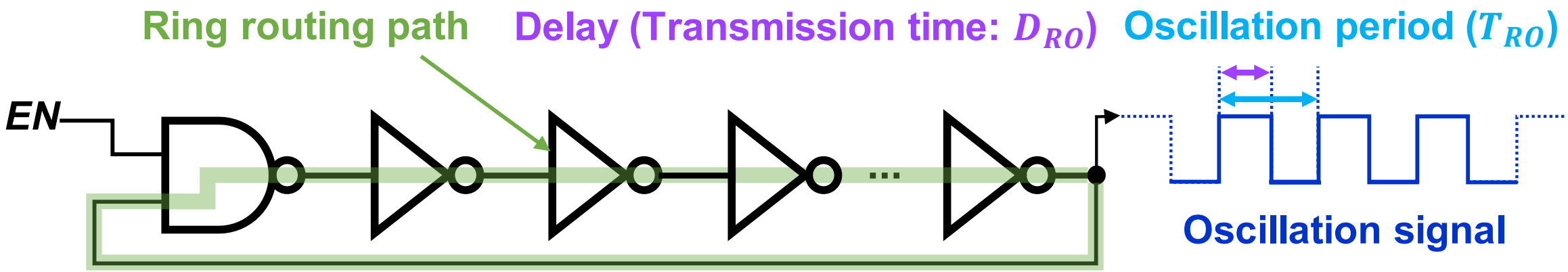
Aging Defect Testing ~ ATD Delay~

- ATD is extremely sensitive to delay variation
- Aging phenomena increase the threshold voltage of the transistors in ATD
  - slow down the switching speed
  - false detection of the address change



Aging Defect Testing ~ Ring oscillator (RO) ~

- **Ring oscillator** is effective way as on-chip digital delay sensor
  - to measure circuit delay variation in a target device (such as in ASIC)

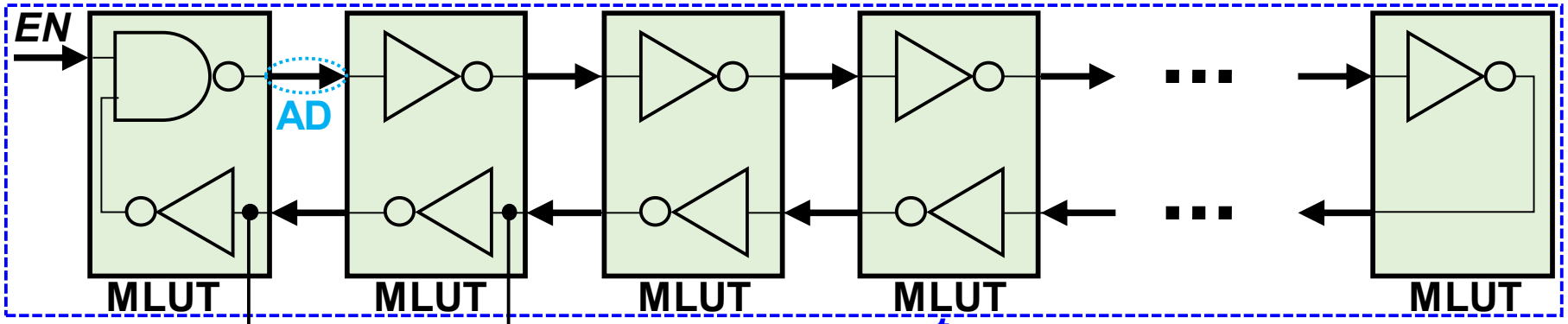


We can calculate Transmission Delay D_{RO} through the oscillation number $N_{osc}^{t_{RO}}$ within a certain oscillation operation time t_{RO} :

$$D_{RO} = \frac{T_{RO}}{2} = \frac{t_{RO}}{2N_{osc}^{t_{RO}}}$$

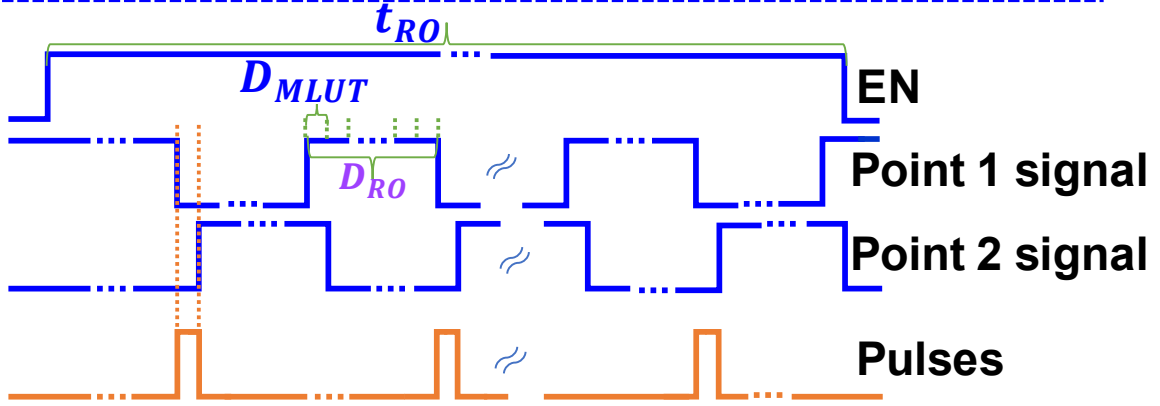
Aging Defect Testing ~ LUT-based Delay-Monitoring ~

Deploy RO in MLUTs



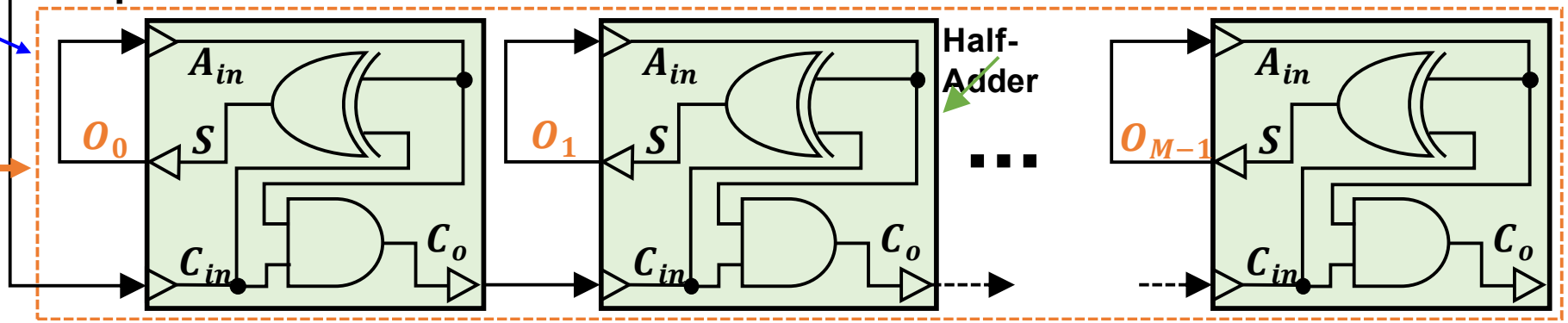
$$D_{MLUT} = \frac{D_{RO}}{N_{AD}} = \frac{t_{RO}}{2N_{osc}^{t_{RO}} N_{AD}}$$

$$N_{osc}^{t_{RO}} = (O_{M-1} \cdots O_1 O_0)_2$$



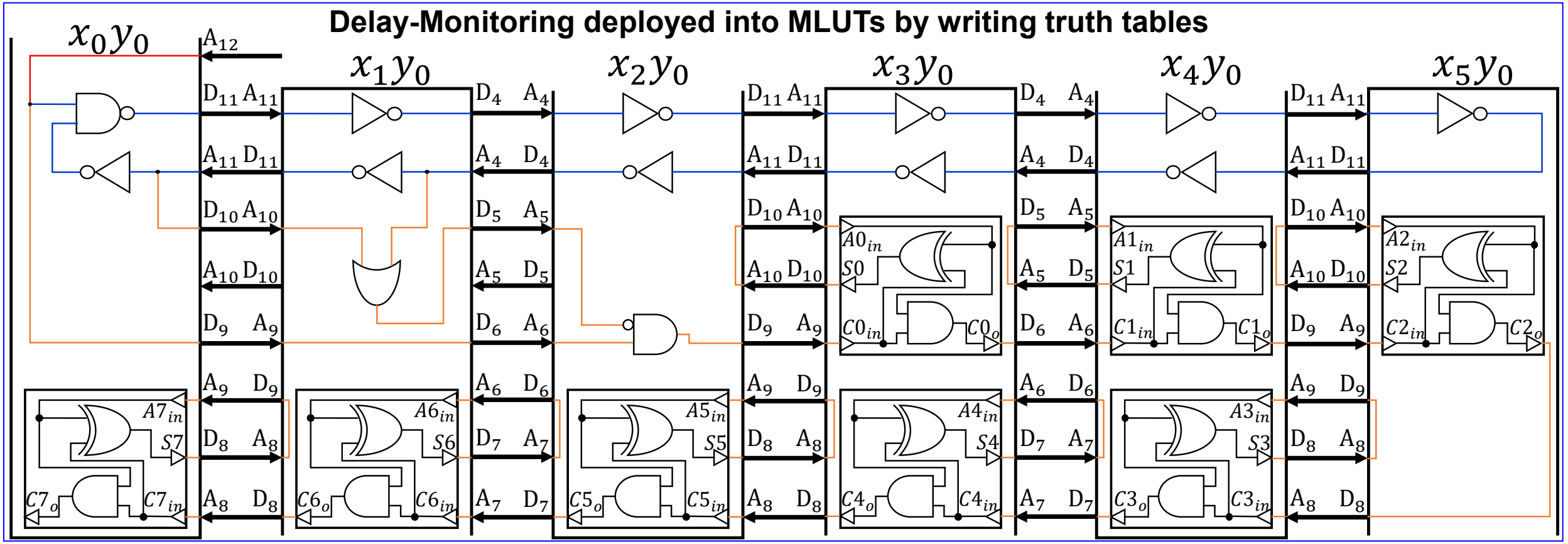
Edge detection pulse

Deploy RO Counter in MLUTs



Aging Defect Testing ~ Simulation for LUT-based Delay-Monitoring ~

- **MPLD Design (6x6 MLUTs array)**: Verilog HDL
- **Delay-Monitoring**: Truth tables



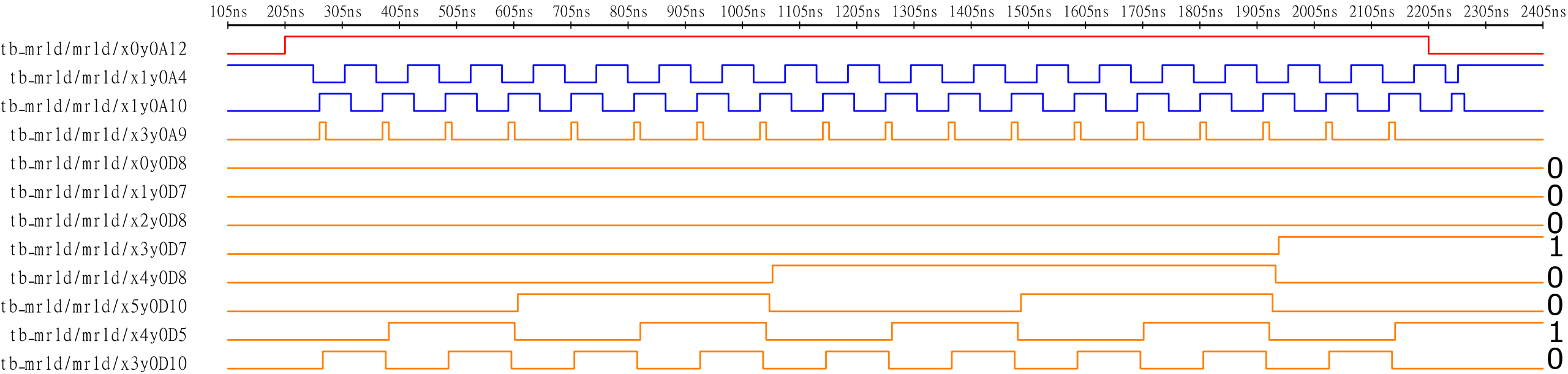
- **Logic simulation experiment using *ModelSim***:

1: route the RO pass through 10 AD interconnects in the measurement area (N_{AD}=10) *(truth tables)*.

2: inject the *5.5ns* delay in the ATD circuit ($D_{ATD} = 5.5\text{ns}$) for each MLUT *(inserting delay)*

3: set the overall oscillation operation time of the RO to *2000ns* (t_{RO}).

Aging Defect Testing ~ Simulation result ~



$$N_{osc}^{t_{RO}} = (00010010)_2 = 18$$

$$D_{MLUT} = \frac{t_{RO}}{2N_{osc}^{t_{RO}} N_{AD}} = \frac{2000ns}{2 \times 18 \times 10} = 5.\dot{5}ns$$

Confirmed

$= D_{ATD} = 5.5\text{ns}$

Outline

- Background
- Objectives
- Reliability Enhancement for Automotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- **Reliability Enhancement for MPLD**
  - **Interconnect Defect Testing**
  - **Aging Monitoring Techniques**
  - **Conclusions**
- Summary
- LIST OF PUBLISHED REFERENCE PAPERS

Conclusions ~ Enhancing the Reliability of MPLD ~

- To guarantee the long-term reliability of the MPLD device, this study proposed
  - interconnect defects test method
    - to identify the interconnect defects under the production phase
  - LUT-based delay monitoring
    - to detect the aging-caused failures in the field

- To evaluate the proposed methods, this study
  - designed an MPLD with a 6×6 MLUTs array
  - performed logic simulations by injecting faults into MPLD
  - confirmed the effectiveness of the proposed methods

Outline

- Background
- Objectives
- Reliability Enhancement for Automotive ECU
  - Multi-Cycle Power-on Self-Test (POST)
  - Test Point Insertion Technique for Multi-Cycle POST
  - Conclusions
- Reliability Enhancement for MPLD
  - Interconnect Defect Testing
  - Aging Monitoring Techniques
  - Conclusions
- **Summary**
- LIST OF PUBLISHED REFERENCE PAPERS

Summary ~ Reliability Enhancement of Edge Computing Devices ~

- **Focuses on enhancing the reliability of two types of edge devices**
  - Automotive ECU device & MPLD (memory-based programmable logic device)

- **For ECU device**
  - Test Point Insertion Technique for Multi-Cycle Power-on Self-Test
    - ✓ to satisfy Functional Safety (ISO 26262)

      test application time (<50ms) & fault coverage (>90%)

Results: 24.4X reduction in scan-in patterns for achieving 90% fault coverage

- **For MPLD device**
  - Interconnect defects test method & Aging Monitoring Techniques
    - ✓ to guarantee the long-term reliability of the MPLD

      manufacturing defects & in-filed aging

Results: Identified all Interconnect defects and detected aging-induced delay

- **Future work**
  - Implement test point insertion technique in industrial design for ECU
  - Design for testability and built-in self-tests for the MPLD
  - Quantitative analysis of aging phenomena for MPLD

LIST OF PUBLISHED REFERENCE PAPERS

Transactions:

1. **Test Point Insertion for Multi-Cycle Power-On Self-Test**
   Senling Wang, Xihong Zhou, Yoshinobu Higami, Hiroshi Takahashi, Hiroyuki Iwata, Yoichi Maeda, Jun Matsushima
   ACM Transactions on Design Automation of Electronic Systems (ACM TODES), Vol. 28, No. 3, pp. 1-21, May 2023. (Peer-reviewed)

2. **Testing and Delay-Monitoring for the High Reliability of Memory-based Programmable Logic Device**
   Xihong Zhou, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi
   IEICE Transactions on Information and Systems (*Minor Revision*)

International Conference Papers:

3. **Diagnosis for Interconnect Faults in Memory-based Reconfigurable Logic Device**
   Xihong Zhou, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi
   22nd IEEE Workshop on RTL and High Level Testing (WRTLT), Nov. 2021 (Peer-reviewed)

4. **MNN: A Solution to Implement Neural Networks into a Memory-based Reconfigurable Logic Device (MRLD)**
   Xihong Zhou, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi, Mitsunori Katsu, Shoichi Sekiguchi
   36th International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC), Jun. 2021. (Peer-reviewed)

5. **Aging Monitoring for Memory-based Reconfigurable Logic Device (MRLD)**
   Xihong Zhou, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi,
   35th International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC), Jul. 2020. (Peer-reviewed)

LIST OF PUBLISHED REFERENCE PAPERS

National Conference Papers :

6. **Implementing Neural Networks on Memory-based Reconfigurable Logic Device (MRLD)**
   Xihong ZHOU, Senling WANG, Yoshinobu HIGAMI, Hiroshi TAKAHASHI, Masayuki SATO, Mitsunori KATSU, Shoichi SEKIGUCHI
   30th Microelectronics Symposium (MES), Sep. 2020.

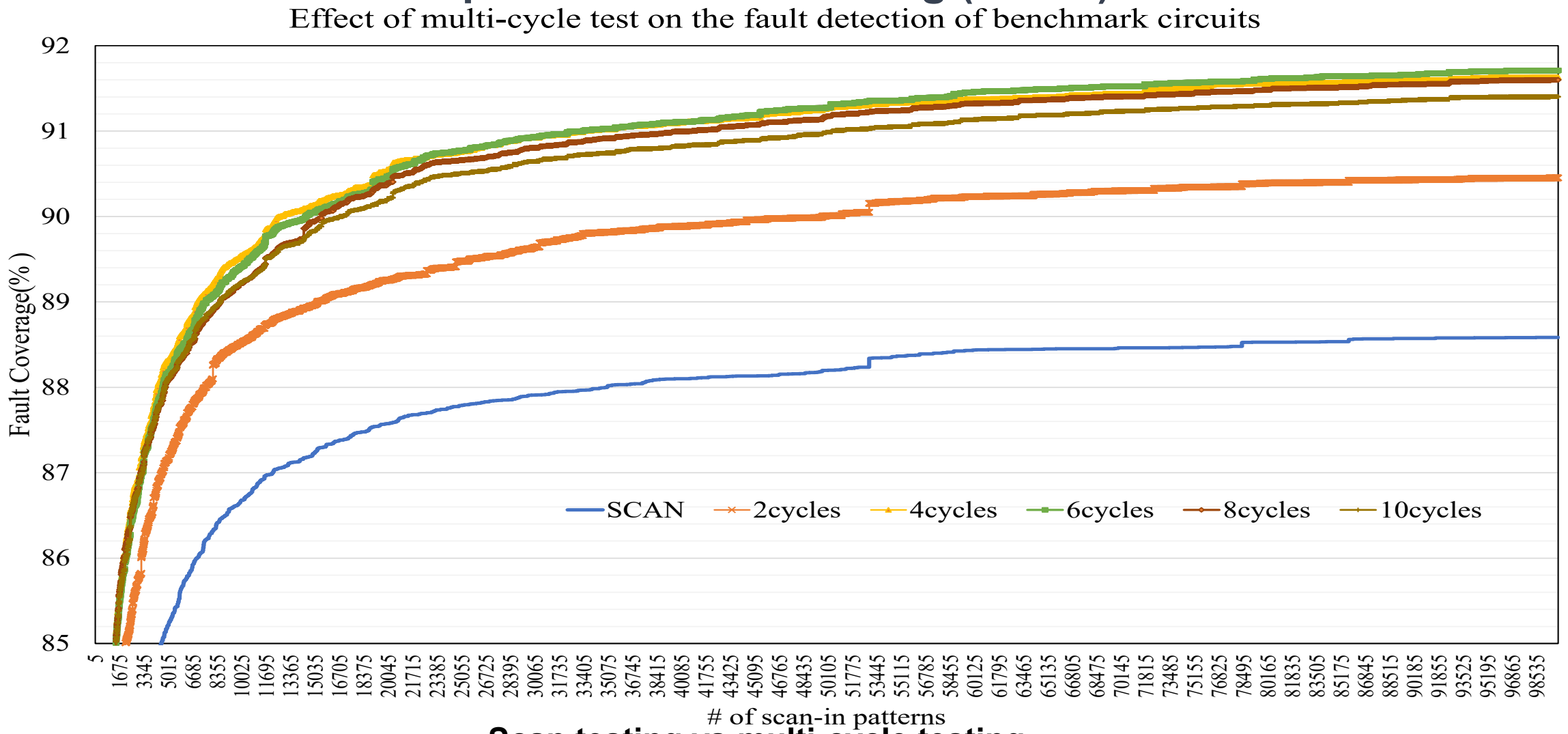
7. メモリベース論理再構成デバイス**(MRLD)**における劣化状態検知のためのリングオシレータ実装
   周 細紅, 王 森レイ, 樋上 喜信, 高橋 寛
   第**34**回エレクトロニクス実装学会春季講演大会講演集 **(JIEP), Mar. 2020.**

8. マルチサイクルテストにおける故障検出強化のためのテストポイント挿入法
   青野 智己, 中岡 典弘, 周 細紅, 王 森レイ, 樋上 喜信, 高橋 寛, 岩田 浩幸, 前田 洋一
   電子情報通信学会技術研究報告 **(IEICE-DC), vol. 119, no. 420, pp. 19-24, Feb. 2020.**

Thanks for your Attention!

Experimental Results ~ Efficiency of the Multi-Cycle Test ~

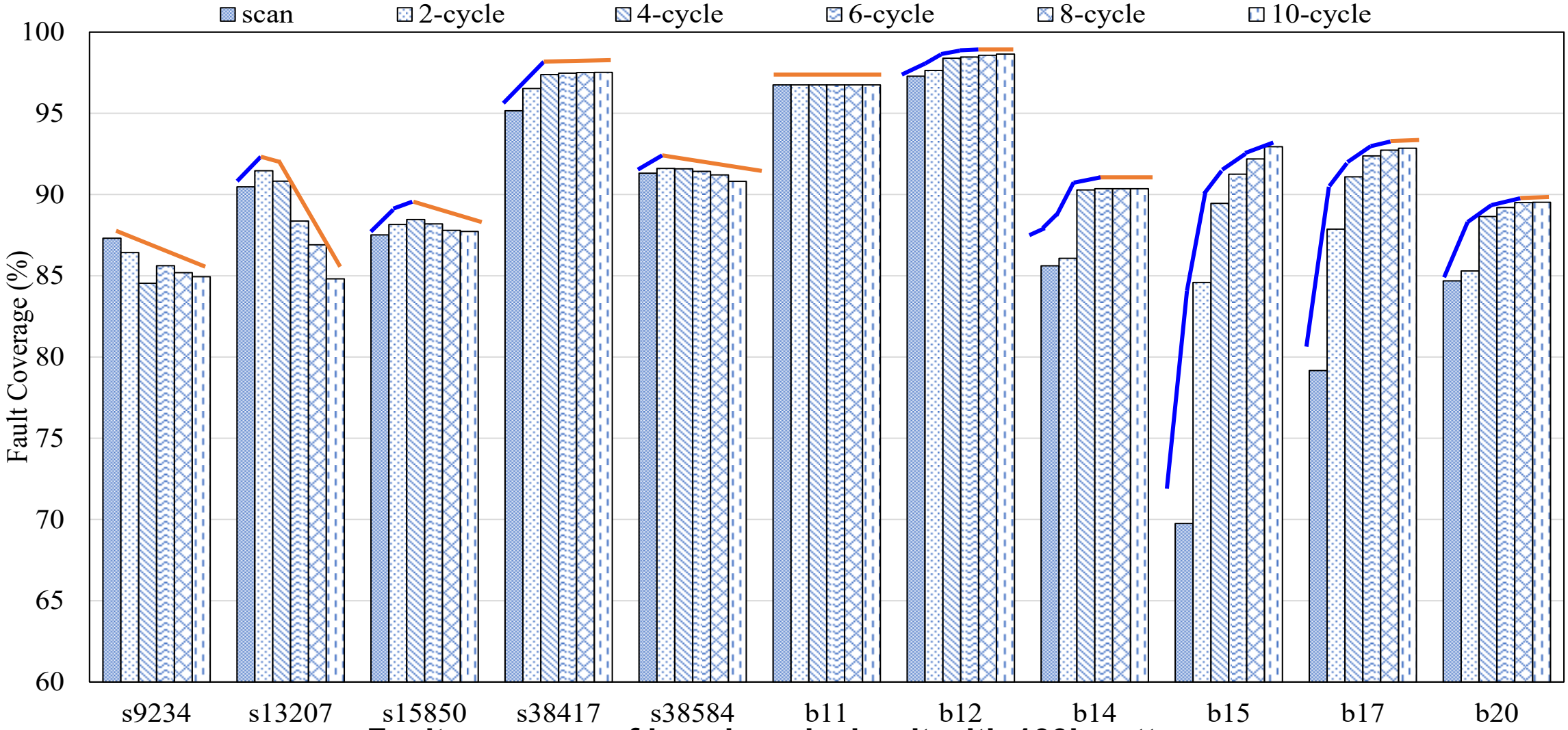
- Multi-cycle test has a statistical improvement in fault detection for most benchmark circuits compared with scan testing (SCAN)



Scan testing vs multi-cycle testing

Experimental Results ~ Efficiency of the Multi-Cycle Test ~

- Multi-cycle test achieved an **increase** in fault coverage at **2, 4 capture cycle**
- **increasing** to **10 cycles**, fault coverage is **slowing down** or **getting degraded**



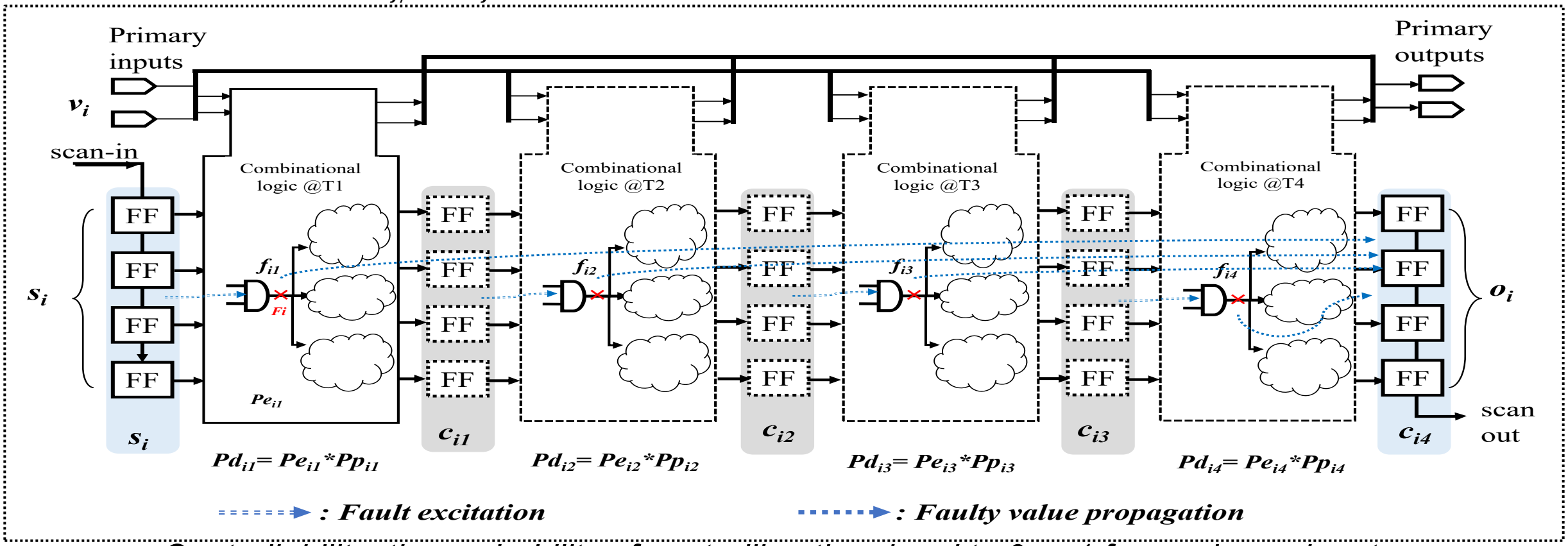
Fault coverage of benchmark circuit with 100k patterns

Fault Detection Model in Multi-Cycle Test

- **Detection probability $Pd_{i/s}$ of a stuck-at-fault $F_{i/s}$ *in a multi-cycle test,* can be estimated by computing the $s^{\neg}$ controllability ($C_{ij/s^{\neg}}$) and the observability (O_{ij}) of a signal line :**

$$Pd_{i/s} = 1 - \prod_{j=1}^{M} \left(1 - C_{ij/s^{\neg}} \times O_{ij}\right) \qquad (4.1)$$

where $C_{ij/s^{\neg}}$, O_{ij} by existing method: COP (controllability observability procedure)



Controllability: the probability of controlling the signal to 0 or 1 from primary inputs
Observability: the probability of observing the signal at primary outputs

Evaluation Metrics for Test Point Selection

- **BD(x): degree of controllability bias at line x** that would impact the fault detection, where **BD(x)>0** denotes a **positive bias**, **BD(x)<0** denotes a **negative bias**.

$$BD(x) = \frac{fg_{x/1} - fg_{x/0}}{M} \sum_{j=1}^{M} (p_{xj/0} - p_{xj/1}) \qquad (5.3)$$

BD(x)<0: should CPI

- **CD(x): degree of contribution to relax the controllability bias** as forcing the 0/1-controllability of line x to 0.5/0.5. **CD(x)>0** denotes a **positive contribution**, **CD(x)<0** denotes a **negative contribution** that would be achieved by CP insertion.

$$CD(x) = \frac{fg_{x/1} - fg_{x/0}}{M} \sum_{j=1}^{M} (0.5 - p_{xj/0}) \qquad (5.4)$$

CD(x)>0: should CPI

- **U: cost function to evaluate the quality** of **CPs** and **OPs**

$$\Delta U = U^{org} - U^{tp} = \frac{1}{|F|} \sum_{\forall i/s \in F} \left(\frac{1}{Pd_{i/s}^{org}} - \frac{1}{Pd_{i/s}^{tp}} \right) \qquad (5.7)$$

Larger ΔU: better TP

use the s27 circuit as an example for BD and CD

$$BD(x)=(p_{x/0}-p_{x/1}) \times (fg_{x/1}-fg_{x/0}) \quad (5.1)$$

$$CD(x)=(p_{x/0}-0.5) \times fg_{x/0}+(p_{x/1}-0.5) \times fg_{x/1}=(0.5-p_{x/0}) \times (fg_{x/1}-fg_{x/0}) \quad (5.2)$$

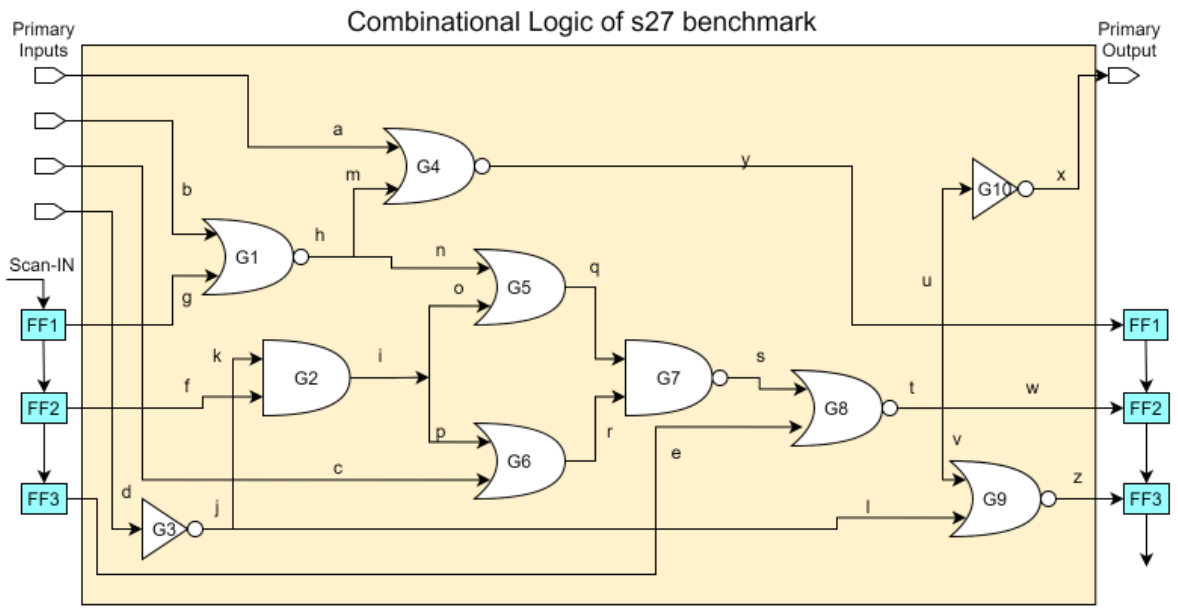
For signal line *i*, two paths connect with the PPO (FF2) through
path 1: $i \rightarrow G5 \rightarrow G7 \rightarrow G8 \rightarrow w$, and
path 2: $i \rightarrow G6 \rightarrow G7 \rightarrow G8 \rightarrow w$.

When the value of *i* is 1, the output of G5, G6, and G7 will be fixed at 1, 1, 0, respectively, thus $fg_{i/1}=3$.
When *i* is 0, the output of G5 and G6 depends on the other input signal lines *n* and *c*, which implies a 0 value at *i* cannot directly cause any fixed gates on the two paths to FFs, thus $fg_{i/0}=0$.

The probability of signal line *i's* values $p_{i/1}$ and $p_{i/0}$ can be calculated using the COP measurement, which is 0.25 and 0.75.

The degree of controllability bias is hereby $BD(i) = 0.5 \times 3 = 1.5$, which represents that the controllability bias at signal line *i* is positive to fault detection.

For the signal line with positive controllability bias, inserting a CP would cause more fixed gates on the fault propagation paths to FFs with a negative contribution to fault detection, e.g., $CD(i)=-0.25 \times 3=-0.75$.



s27 circuit

Evaluation metrics of signal lines in s27

line #	$fg_{x/0}$	$fg_{x/1}$	$p_{x/0}$	$p_{x/1}$	*BD*	*CD*
h	0	2	0.75	0.25	1	-0.5
i	0	3	0.75	0.25	1.5	-0.75
n	0	1	0.75	0.25	0.5	-0.25
q	3	0	0.56	0.44	-0.36	0.18
s	0	2	0.27	0.73	-0.92	0.46

Detailed results of the final fault coverage achieved

- The final fault coverage reached by 100K scan-in patterns

Table 5.3 The final fault coverage reached by 100K scan-in patterns

Circuit	Design for Testability Approaches											
	SCAN	10-Cycle	OPI_ONLY	FullOB	# of CP<1% of gates				# of CP<5% of FFs			
					# of CPs	CPI-ONLY	CPI&OPI	CP&FullOB	# of CPs	CPI-ONLY	CPI&OPI	CP&FullOB
s9234	87.31	84.94	89.94	90.00	55	82.69	89.68	91.80	11	83.3	87.96	88.02
s13207	90.47	84.81	92.20	92.96	79	86.16	92.75	93.89	33	85.6	90.1	91.01
s15850	87.51	87.73	88.48	90.18	104	85.09	87.41	91.52	29	86.47	87.71	90.77
s38417	95.16	97.52	97.96	98.03	141	98.19	98.66	98.72	85	98.00	98.55	98.62
s38584	91.31	90.81	91.59	92.07	97	91.16	91.70	92.28	72	90.27	90.93	91.53
b11	96.75	96.75	96.75	96.75	2	98.03	98.03	98.03	1	96.82	96.82	96.82
b12	97.28	98.64	98.68	98.68	9	99.18	99.21	99.21	6	97.6	97.6	97.64
b14	85.61	90.36	90.38	90.40	44	93.71	93.96	94.07	12	93.81	94.04	94.08
b15	69.75	92.94	92.95	92.95	8	98.35	98.36	98.36	8	98.35	98.36	98.36
b17	79.17	92.85	92.85	92.86	201	97.76	97.81	97.83	70	96.27	96.29	96.32
b20	84.69	89.52	89.66	89.69	88	93.10	93.61	93.94	24	92.68	93.17	93.22

Detailed results of the final fault coverage achieved

- The number of scan-in patterns to achieve 90% fault coverage

Table 5.4 The number of scan-in patterns to achieve 90% fault coverage

Circuit	Design for Testability Approaches											
	SCAN	10-Cycle	OPI_ONLY	FullOB	# of CP<1% of total gates				# of CP<5% of FFs			
					# of CPs	CPI-ONLY	CPI&OPI	CP&FullOB	# of CPs	CPI-ONLY	CPI&OPI	CP&FullOB
s9234	>100K	>100K	>100K	>100K	55	>100K	>100K	9180	11	>100K	>100K	>100K
s13207	20560	>100K	11565	7375	79	>100K	6050	4175	33	>100K	59835	8885
s15850	>100K	>100K	>100K	68905	104	>100K	>100K	2380	29	>100K	>100K	4710
s38417	5780	1710	590	460	141	250	80	55	85	310	85	60
s38584	8180	10645	3700	1960	97	1555	575	305	72	12795	960	330
b11	475	120	120	120	2	45	40	35	1	115	115	105
b12	1280	175	170	170	9	100	45	45	6	260	210	195
b14	>100K	58280	58280	53425	44	1285	870	770	12	885	675	605
b15	>100K	4180	4180	4115	8	285	230	170	8	285	230	170
b17	>100K	4305	4300	4300	201	180	130	100	70	260	185	140
b20	>100K	>100K	>100K	>100K	88	4330	2045	935	24	7480	3740	3685

Detailed information of benchmark circuits

- ISCAS89 and ITC99

Table 5.2 Detailed information of benchmark circuits

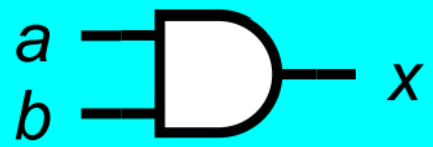
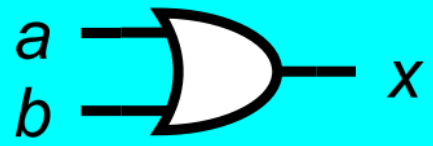
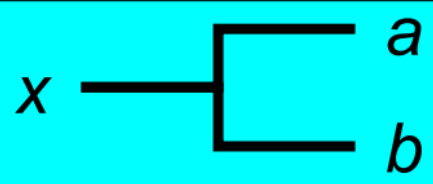
Circuit	# gate	# FF	# of stuck-at fault	N_{cp} (<1% of gates)	N_{cp} (<5% of FFs)	#OPs (FDS-FFs) (<20% of FFs)
s9234	5597	228	6927	55	11	45
s13207	7951	669	9815	79	33	133
s15850	9772	597	11725	104	29	120
s38417	22179	1636	31180	1141	85	327
s38584	19253	1452	36303	97	72	290
b11	437	31	1322	2	1	6
b12	904	121	2797	9	6	24
b14	4444	245	12811	44	12	49
b15	8338	449	23528	8	8	89
b17	22645	1415	65464	201	70	283
b20	8875	490	25338	88	24	98

Controllability/Observability Program [Brglez 1984]

COP

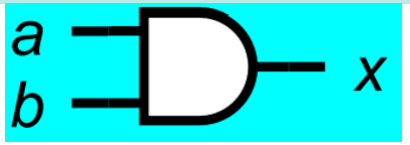
- *Signal probability* of *x* = probability of *x* being logic 1
  - Actual signal probability requires exhaustive simulation
  - Hard to obtain in practice
- COP = Controllability/Observability Program [Brglez 84]
  - Fast algorithm to estimate signal probability
  - C_x = estimated prob($x = 1$)
  - $1\text{-}C_x$ = estimated prob($x = 0$)
  - O_x = estimated probability of *fault effect* in *x* being observed at PO
- C_x and O_x are numbers between 0 and 1
  - *Larger number* means *easier* to control or observe
- Assumptions
  - 1. Ignore fanout reconvergence for fast run time
  - 2. PI are independent random numbers: ½ zero and ½ one

Controllability/Observability Program [Brglez 1984]

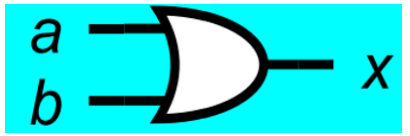
	C_x	O_a
x = PI	*0.5*	
x = PO		*0.5*
	$C_x = C_a \times C_b$	$O_a = O_x \times C_b$
	$C_x = 1 - (1- C_a) \times (1- C_b)$	$O_a = O_x \times (1- C_b)$
	$C_x = C_a = C_b$	$O_x = 1 - (1- O_a) \times (1- O_b)$

Example – Controllability

- **Calculate from PI to PO**

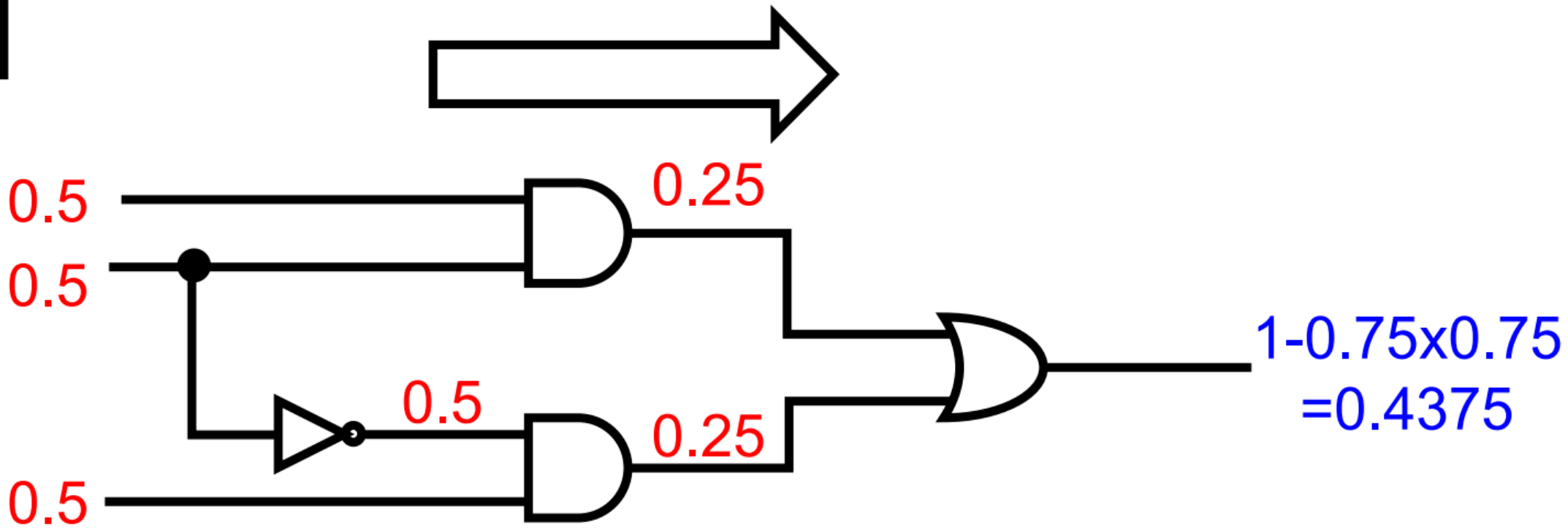


$C_x = C_a \times C_b$

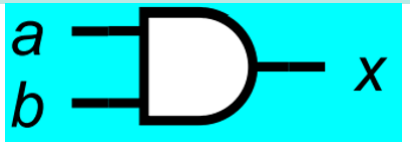


$C_x = 1 - (1 - C_a) \times (1 - C_b)$

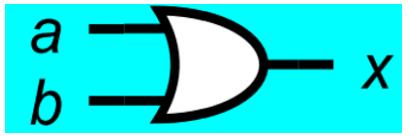
COP C_x



Example – Observability

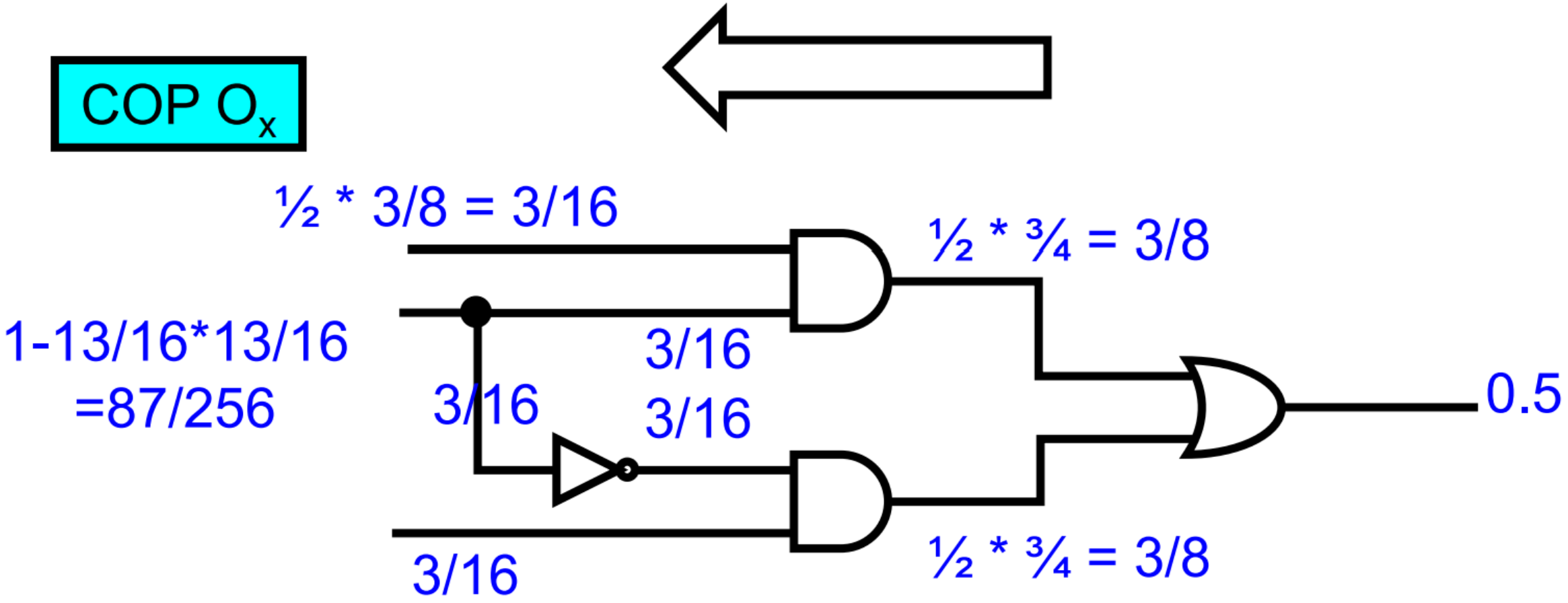


$O_a = O_x \times C_b$



$O_a = O_x \times (1 - C_b)$

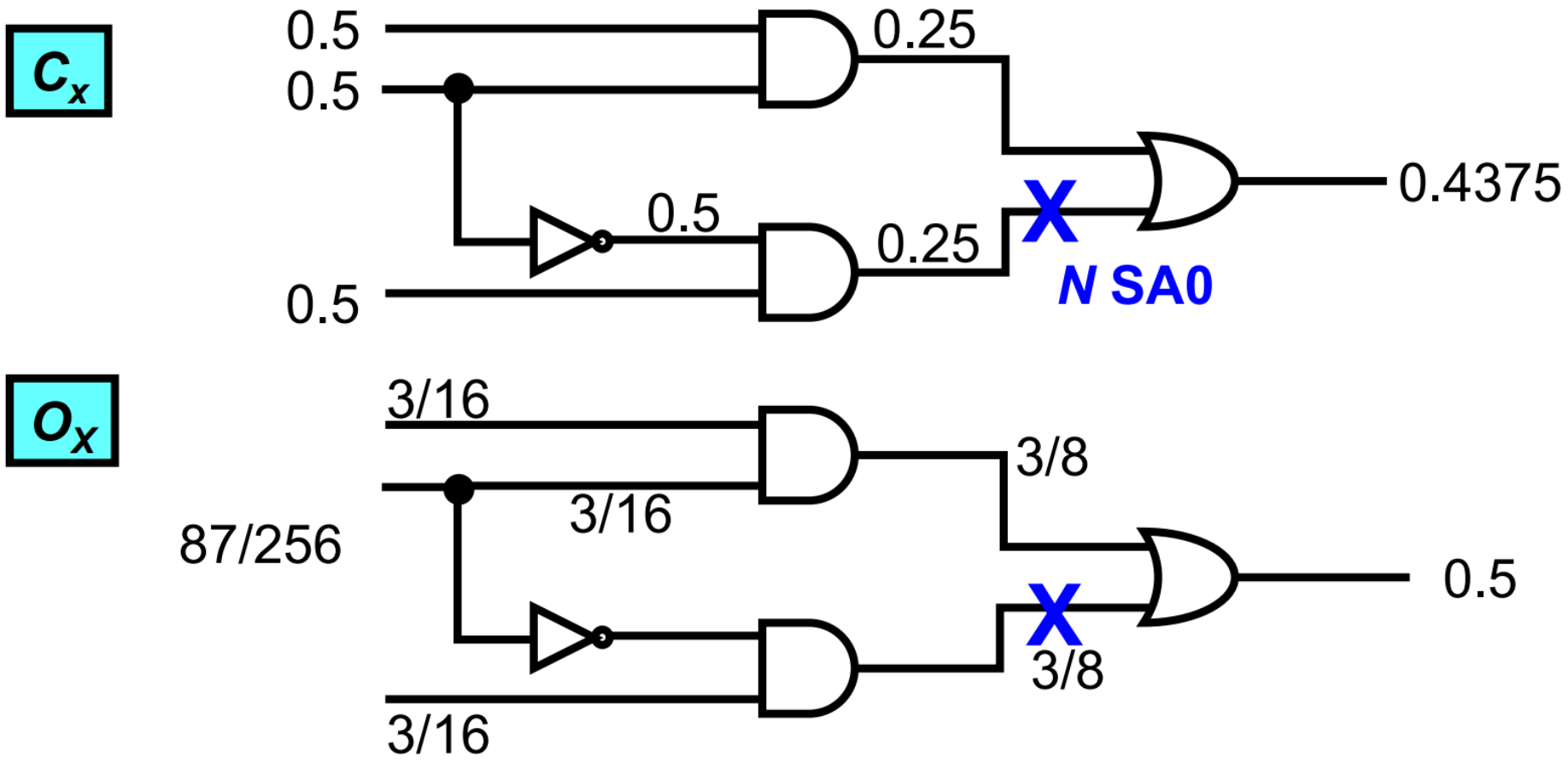
- **Calculate from PO to PI**



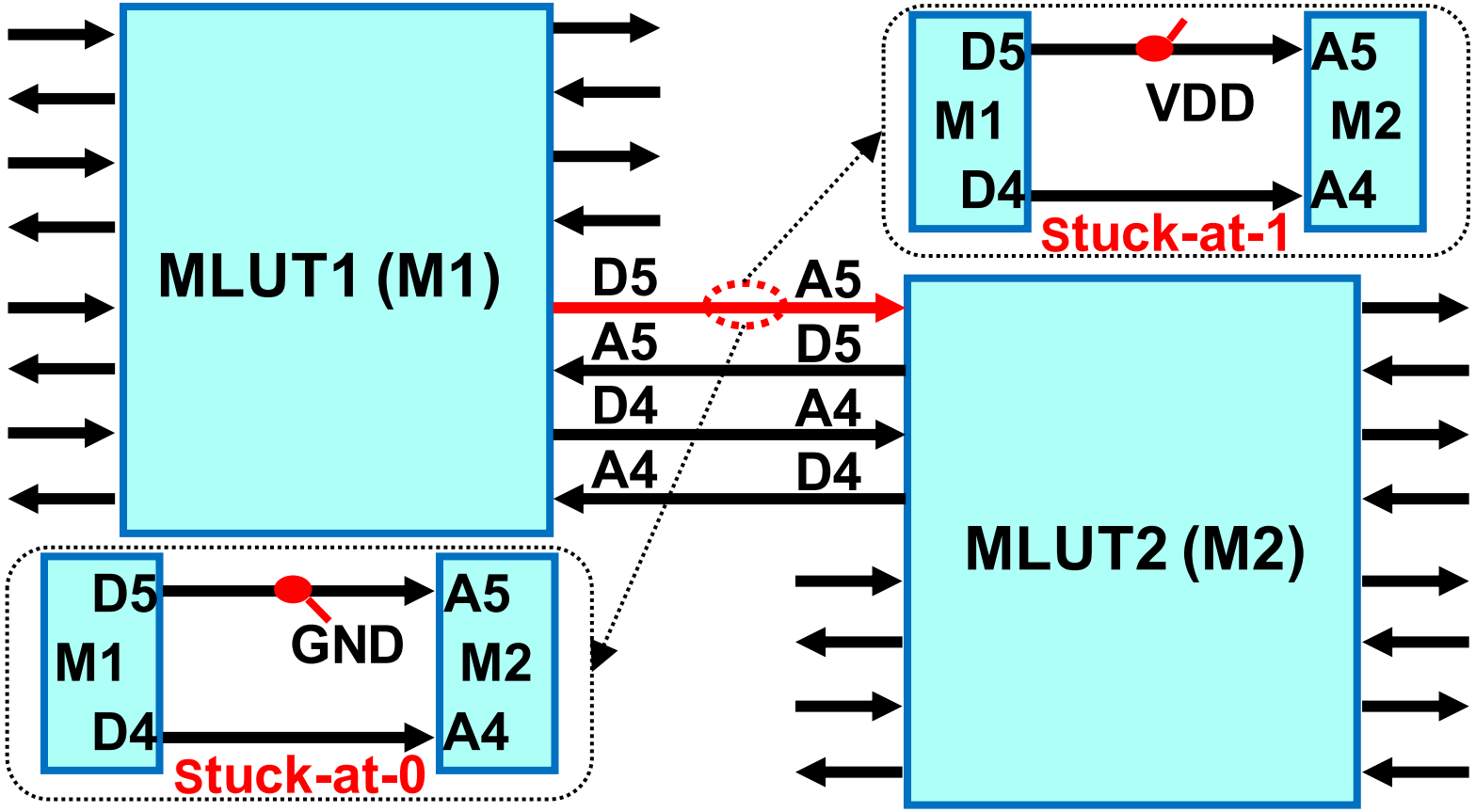
Detection Probability, Pd

- **Pd_f = Probability of detecting a fault f**
  - ***$Pd_{N/SA0}$ = C_N x O_N***
  - ***$Pd_{N/SA1}$ = $(1-C_N)$ x O_N***
- **Larger Pd_f means easier to detect fault f**

Example: $Pd_{N/SA0}$ = 1/4 x 3/8 =3/32



behavior of stuck-at interconnect faults



(a) stuck-at faults.

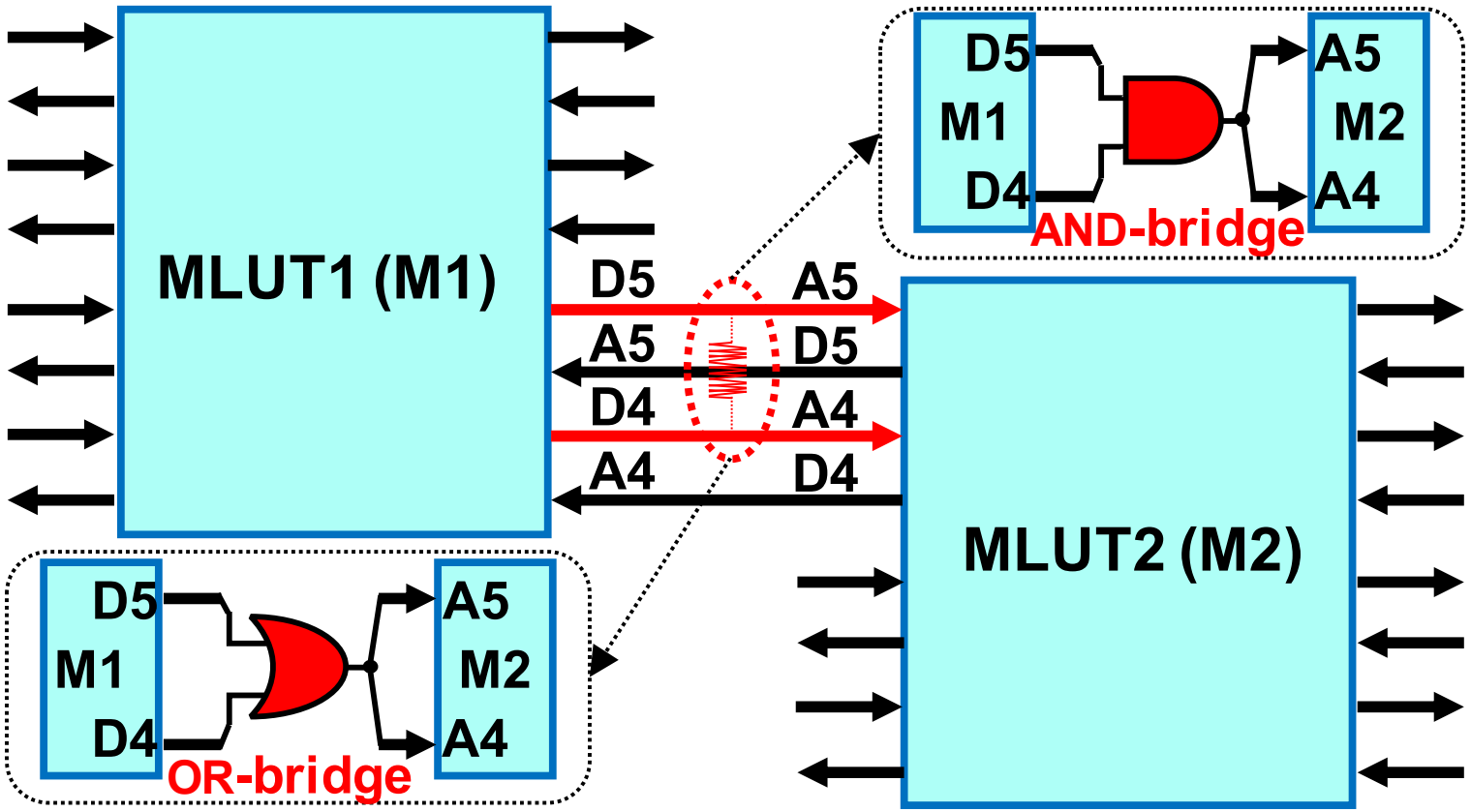
Logic behavior of stuck-at-1

M1D5	M2A5
0	0/1
1	1

Logic behavior of stuck-at-0

M1D5	M2A5
0	0
1	1/0

behavior of bridge interconnect faults



(b) bridge faults.

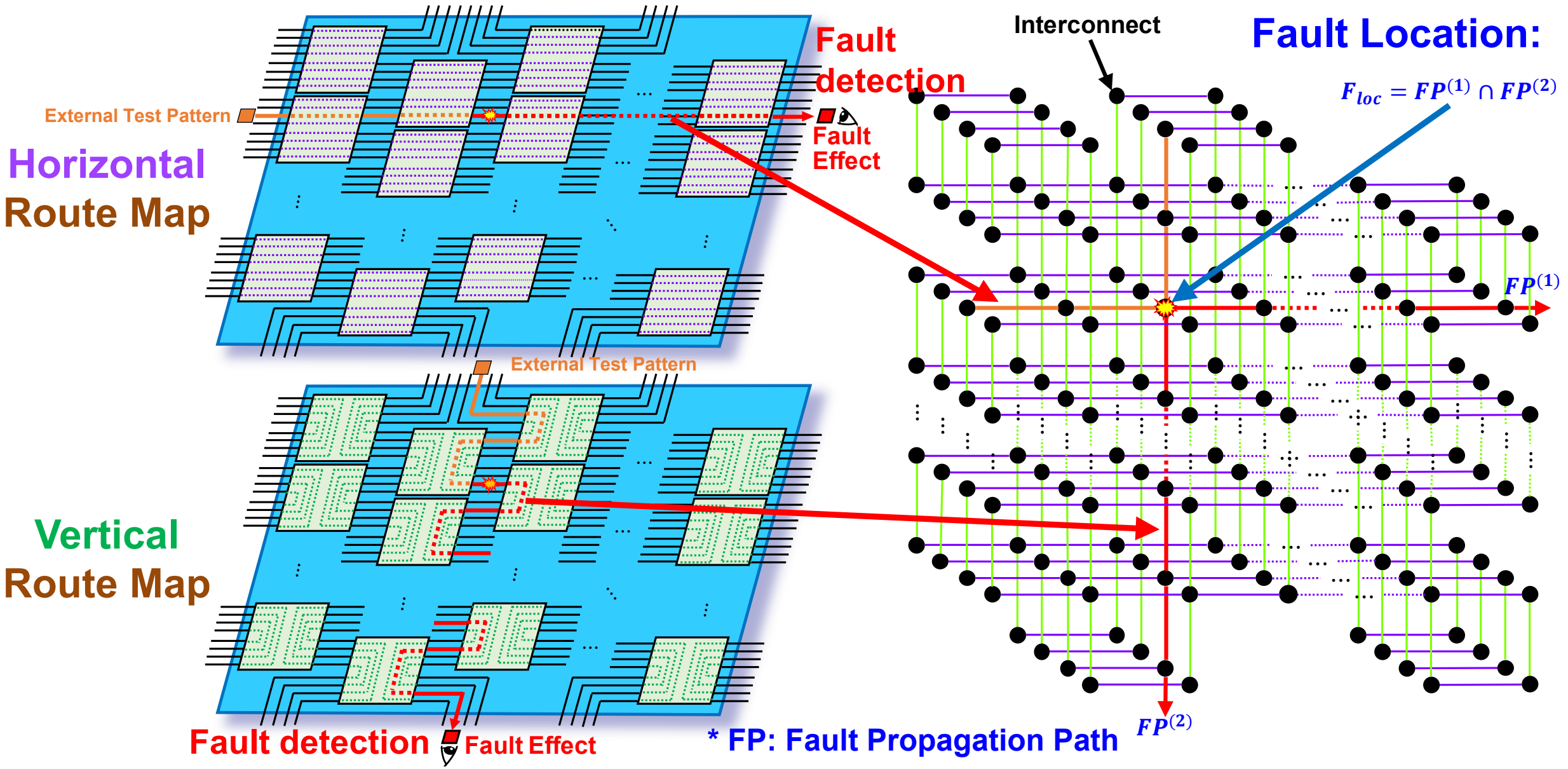
Logic behavior of AND-bridge

M1D5	M1D4	M2A5	M2A4
0	0	0	0
0	1	0	1/0
1	0	1/0	0
1	1	1	1

Logic behavior of OR-bridge

M1D5	M1D4	M2A5	M2A4
0	0	0	0
0	1	0/1	1
1	0	1	0/1
1	1	1	1

Manufacturing Defect Testing ~ Basic Idea ~



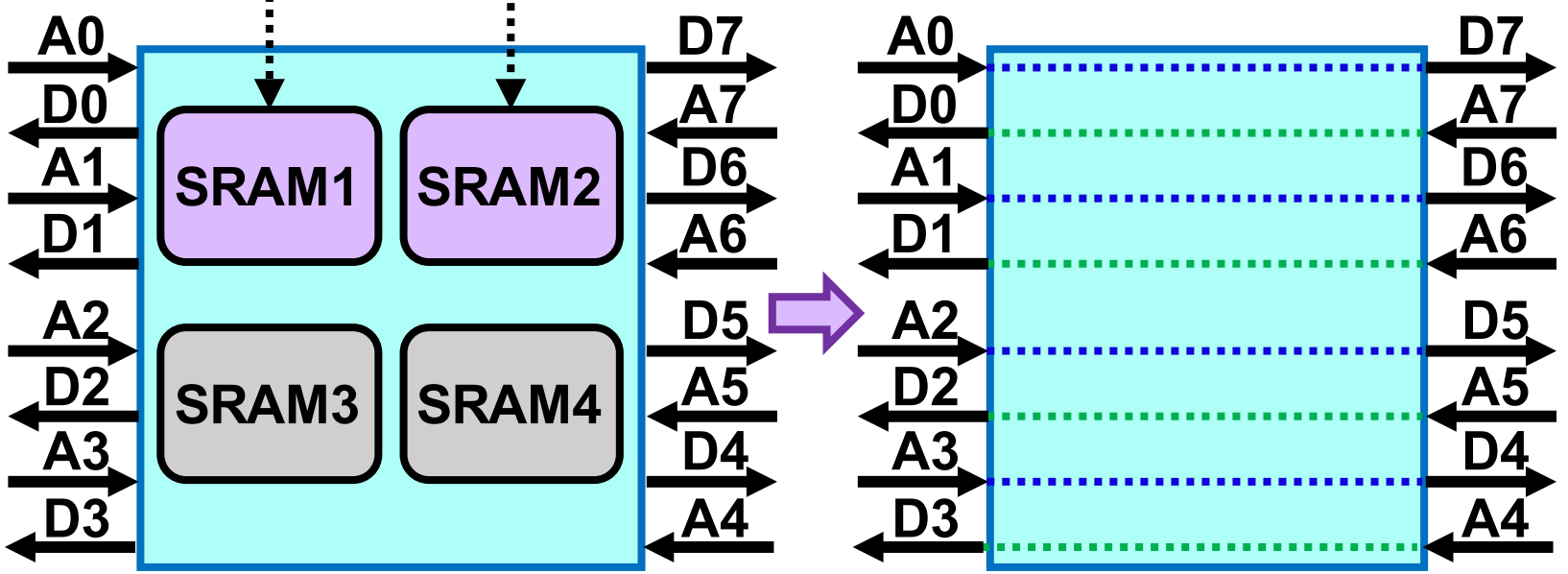
Example of test cubes in SRAMs of an MLUT

Truth table1

Address				Data				
A3	A2	A1	A0	D7	D6	D5	D4	D3~D0
0	0	0	0	0	0	0	0	
0	0	0	1	1	0	0	0	
0	0	1	0	0	1	0	0	*0*
*	*	*	*	*	*	*	*	
1	1	1	1	1	1	1	1	

Truth table2

Address				Data				
A7	A6	A5	A4	D7~D4	D3	D2	D1	D0
0	0	0	0		0	0	0	0
0	0	0	1		1	0	0	0
0	0	1	0	*0*	0	1	0	0
*	*	*	*		*	*	*	*
1	1	1	1		1	1	1	1



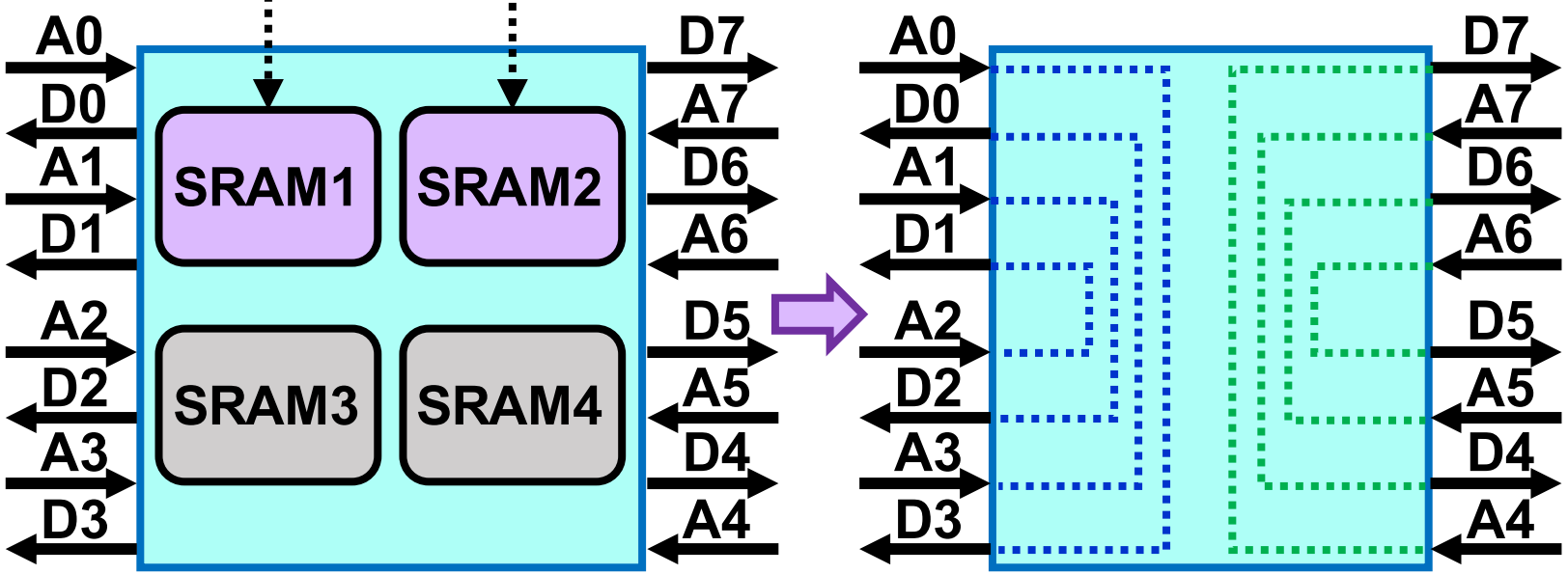
Example of test cubes in SRAMs of an MLUT

Truth table1

Address				Data				
A3	A2	A1	A0	D7~D4	D3	D2	D1	D0
0	0	0	0		0	0	0	0
0	0	0	1		1	0	0	0
0	0	1	0	*0*	0	1	0	0
*	*	*	*		*	*	*	*
1	1	1	1		1	1	1	1

Truth table2

Address				Data				
A7	A6	A5	A4	D7	D6	D5	D4	D3~D0
0	0	0	0	0	0	0	0	
0	0	0	1	1	0	0	0	
0	0	1	0	0	1	0	0	*0*
*	*	*	*	*	*	*	*	
1	1	1	1	1	1	1	1	



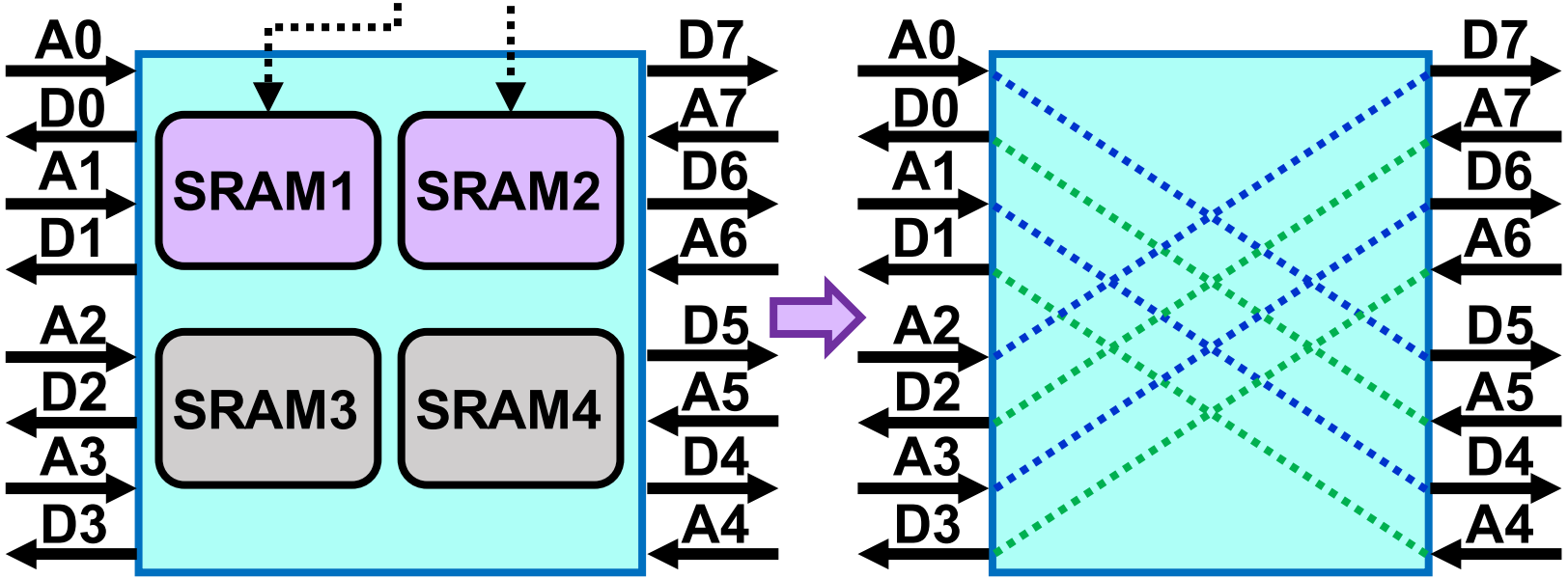
Example of test cubes in SRAMs of an MLUT

Truth table1

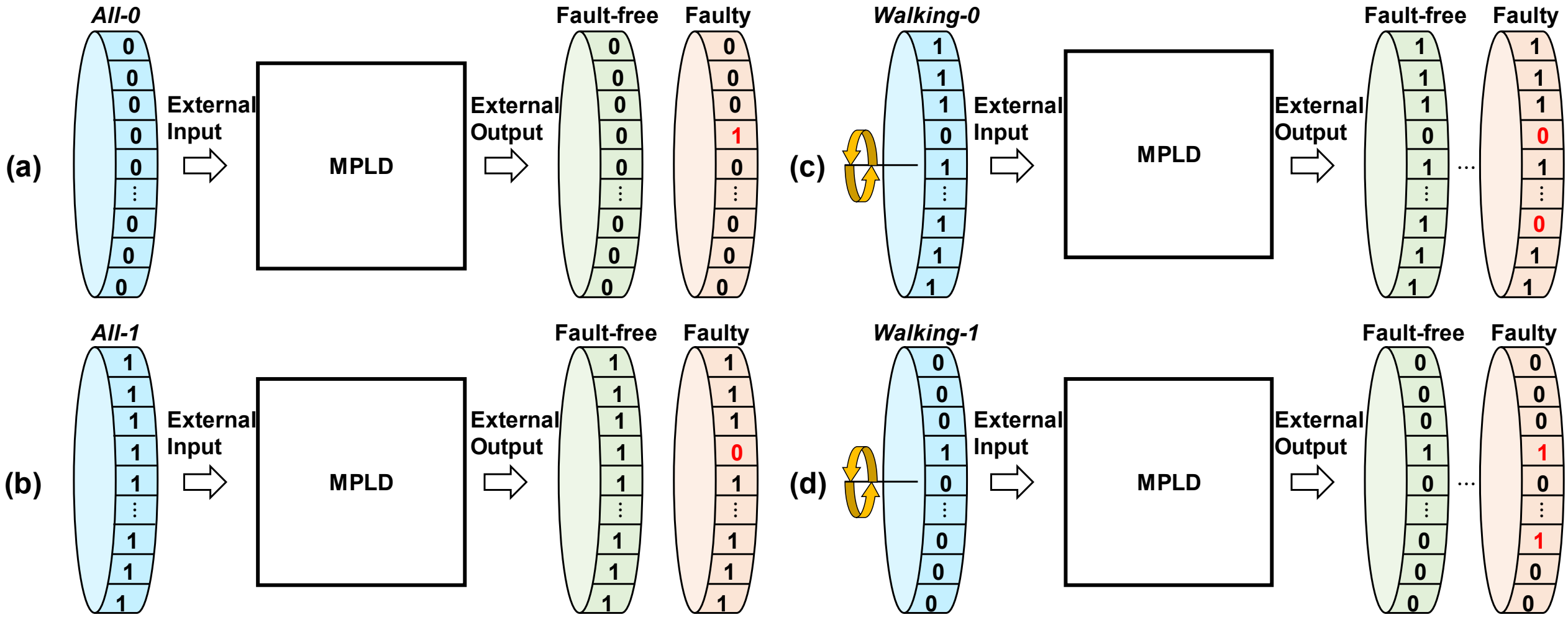
Address				Data				
A3	A2	A1	A0	D7	D6	D5	D4	D3~D0
0	0	0	0	0	0	0	0	
0	0	0	1	0	0	1	0	
0	0	1	0	0	0	0	1	*0*
*	*	*	*	*	*	*	*	
1	1	1	1	1	1	1	1	

Truth table2

Address				Data				
A7	A6	A5	A4	D7~D4	D3	D2	D1	D0
0	0	0	0		0	0	0	0
0	0	0	1		0	0	1	0
0	0	1	0	*0*	0	0	0	1
*	*	*	*		*	*	*	*
1	1	1	1		1	1	1	1



Applying mechanisms of external patterns



Manufacturing Defect Testing ~ Testing Procedure~

Definitions:

- N_{rm}: *number of route maps.*
- rm_i: *route map i;* $i \in [1, N_{rm}]$.
- $TC^{(i)}$: *test cubes creating* rm_i.
- $N_{FE}^{(i)}$: *number of observed fault effects under* rm_i.
- $FP_k^{(i)}$: *fault propagation path k obtained under* rm_i; $k \in [1, N_{FE}^{(i)}]$.
- $FP^{(i)}$: *fault propagation path set under* rm_i.
- F_{loc}: *fault location.*

Process:

(1) *Test under* rm_i *for* $i \in [1, N_{rm}]$:

(a) Configure $TC^{(i)}$ into each MLUT to create rm_i.

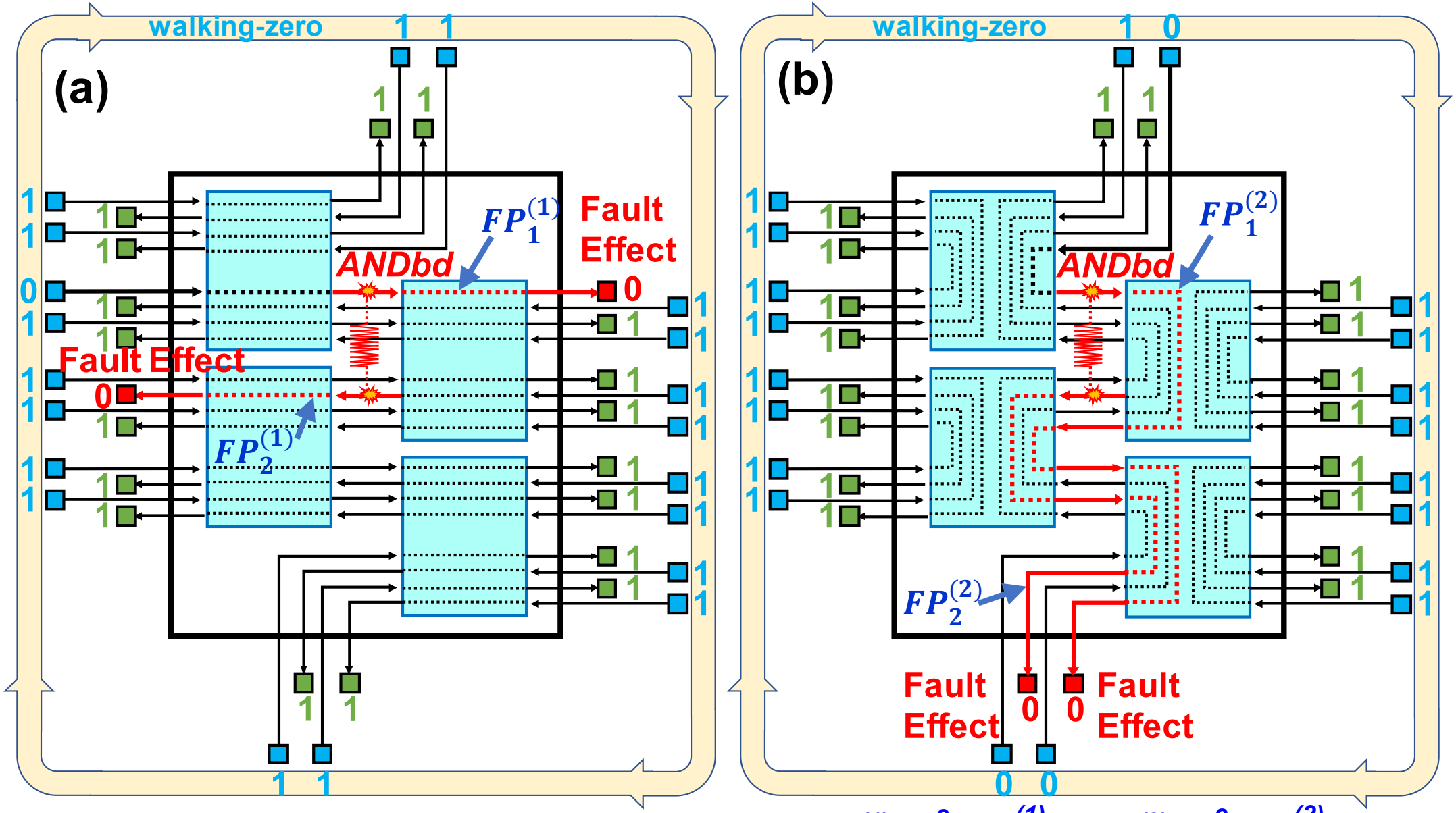
(b) Apply external test patterns to the input ports of MPLD.

(c) Observe fault effects. If $N_{FE}^{(1)}=0$, end testing (fault-free).

(d) Obtain the fault propagation path set: $FP^{(i)} = \bigcup_{k=1}^{N_{FE}^{(i)}} FP_k^{(i)}$.

(2) *Identify fault location*: $F_{loc} = \bigcap_{i=1}^{N_{rm}} FP^{(i)}$.

Manufacturing Defect Testing ~ example ~ --- AND-bridge fault



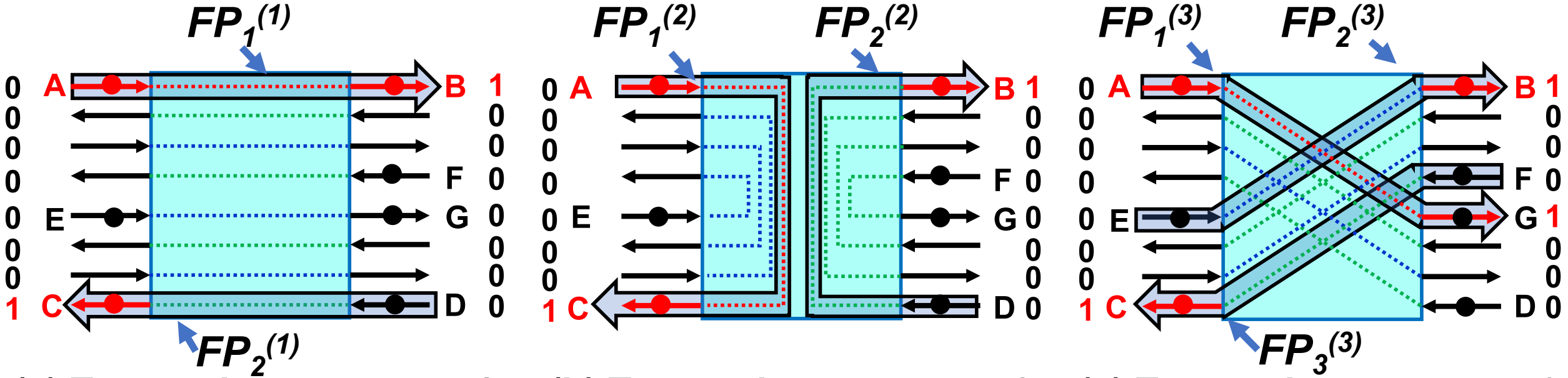
$F_{loc}=FP^{(1)} \cap FP^{(2)}$

$(FP^{(1)}=\cup_{k=1}^{2} FP_k^{(1)},\quad FP^{(2)}=\cup_{k=1}^{2} FP_k^{(2)})$

Implementation Procedure

Step 1: select measurement area (MLUTs);
Step 2: deploy RO and counter;
Step 3: create the truth tables for each MLUT in the area;
Step 4: write the truth tables into corresponding MLUTs;
Step 5: set the MPLD to logic operation mode;
Step 6: set oscillation operation time (EN=1);
Step 7: observe the oscillation number (counter outputs).

Example to test multiple faults



(a) Test under route map 1 (b) Test under route map 2 (c) Test under route map 3

$$FP^{(1)} = \cup_{k=1}^{N_{FE}^{(1)}} FP_k^{(1)} = FP_1^{(1)} \cup FP_2^{(1)} = \{A,B\} \cup \{C,D\} = \{A,B,C,D\},$$

$$FP^{(2)} = \cup_{k=1}^{N_{FE}^{(2)}} FP_k^{(2)} = FP_1^{(2)} \cup FP_2^{(2)} = \{A,C\} \cup \{D,B\} = \{A,B,C,D\},$$

$$\cap_{i=1}^{2} FP^{(i)} = FP^{(1)} \cap FP^{(2)} = \{A,B,C,D\}.$$

$$FP^{(3)} = \cup_{k=1}^{N_{FE}^{(3)}} FP_k^{(3)} = FP_1^{(3)} \cup FP_2^{(3)} \cup FP_3^{(3)} = \{A,G\} \cup \{E,B\} \cup \{F,C\} = \{A,B,C,E,F,G\}.$$

$$\cap_{i=1}^{3} FP^{(i)} = FP^{(1)} \cap FP^{(2)} \cap FP^{(3)} = \{A,B,C,D\} \cap \{A,B,C,E,F,G\} = \{A,B,C\}.$$