Logic Built in Self Test

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,

IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

Introduction

- The complexity of circuits is increasing day by day.
- So testing the circuit from input, output pads becomes laborious and time consuming.
- Hence there is a necessity to reduce the complexity of testing.
- For this purpose Built-In Self-Test (BIST) is developed.

BIST

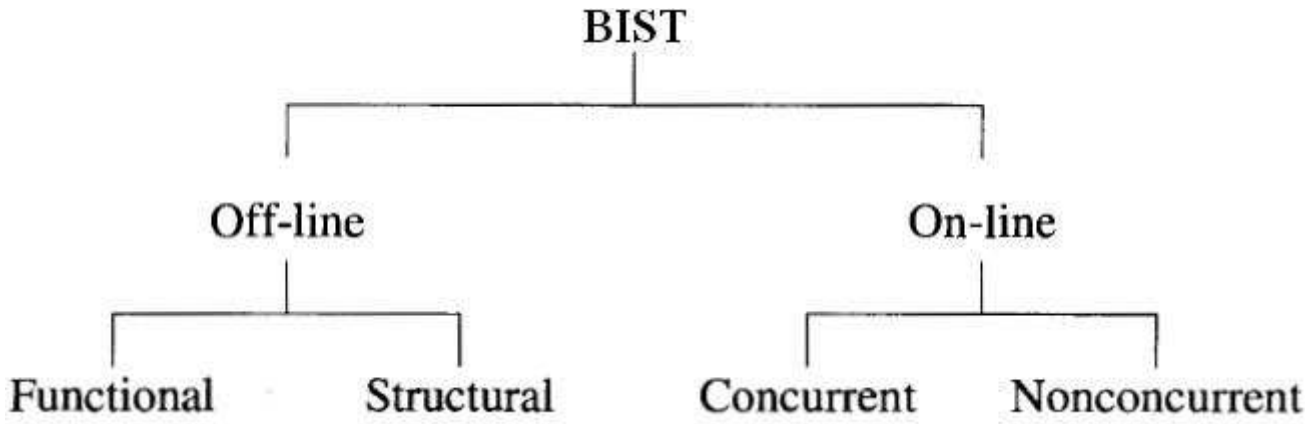
- It is a design technique in which additional circuits are added to the functional blocks which enable testing of the circuit by itself.
- It is a combination of the concepts of built-in test and self-test.
- By using the concept of BIST, there is a great reduction in the testing time.

Logic Built-In Self-Test

- TPG
  - Constructed from linear feedback shift register (LFSR) or cellular automata
  - Exhaustive testing – all possible 2^n test patterns
  - Pseudo-random testing – a subset of 2^n test patterns
  - Pseudo-exhaustive testing
- ORA
  - Constructed from multiple-input signature register (MISR)

Types of BIST

- There are broadly two methods of carrying BIST:
  - Online BIST
    - Concurrent BIST
    - Non-concurrent BIST
  - Offline BIST
    - Functional BIST
    - Structural BIST



Online BIST

- In **online BIST**, the circuit may be made to test itself without being disconnected from the system.

- **Concurrent online BIST**: Allowing the testing to be carried out while the circuit performs its normal operation.
- **Non-concurrent online BIST**: Normal operation is put off and only the testing would be carried out. The test process can be interrupted any time so that the normal operation can resume.

Offline BIST

- In **offline BIST**, the circuit is disconnected from its normal operation and testing is carried out.
- It is applicable at manufacturing, field, operational levels.
- It uses test pattern generators (TPG) and output response analyzers (ORA).
- It cannot detect errors at first occurrence which is possible with many online BIST techniques.

Types of Offline BIST

- **Functional offline BIST**: The execution of the test is based on the functional description of circuit under test (CUT) and employs a functional fault model.

- **Structural offline BIST:** The execution of the test is based on the structure of the CUT which uses explicitly structural fault models.

Requirements for BIST process

For any self testing process, the following are necessary:

- Circuit under test (CUT),
- Test pattern generator (TPG),
- Output response analyzer (ORA),
- A distribution system (DIST) to transmit data from TPG to CUT and from CUT to ORA.
- A BIST controller for controlling the BIST circuitry and the CUT during self test.

BIST Architectures

The BIST architectures are:

- Embedded architectures
- Separate architectures

- In embedded architectures, the registers used for TPGs and ORAs are integral part of the functional circuit.
- In separate architectures, the TPGs and ORAs are separately provided which will come into picture while testing.

BIST Architectures Classification

Based on the number of TPGs and ORAs that are provided for carrying out BIST operation, they are classified as:

- Centralized BIST architectures – Several CUTs share TPG and ORA circuitry. This leads to reduced overhead but increased test time.
- Distributed BIST architectures – Each CUT is associated with its own TPG and ORA. This leads to more overhead but less time and usually more accurate diagnosis.

Logic BIST Architectures

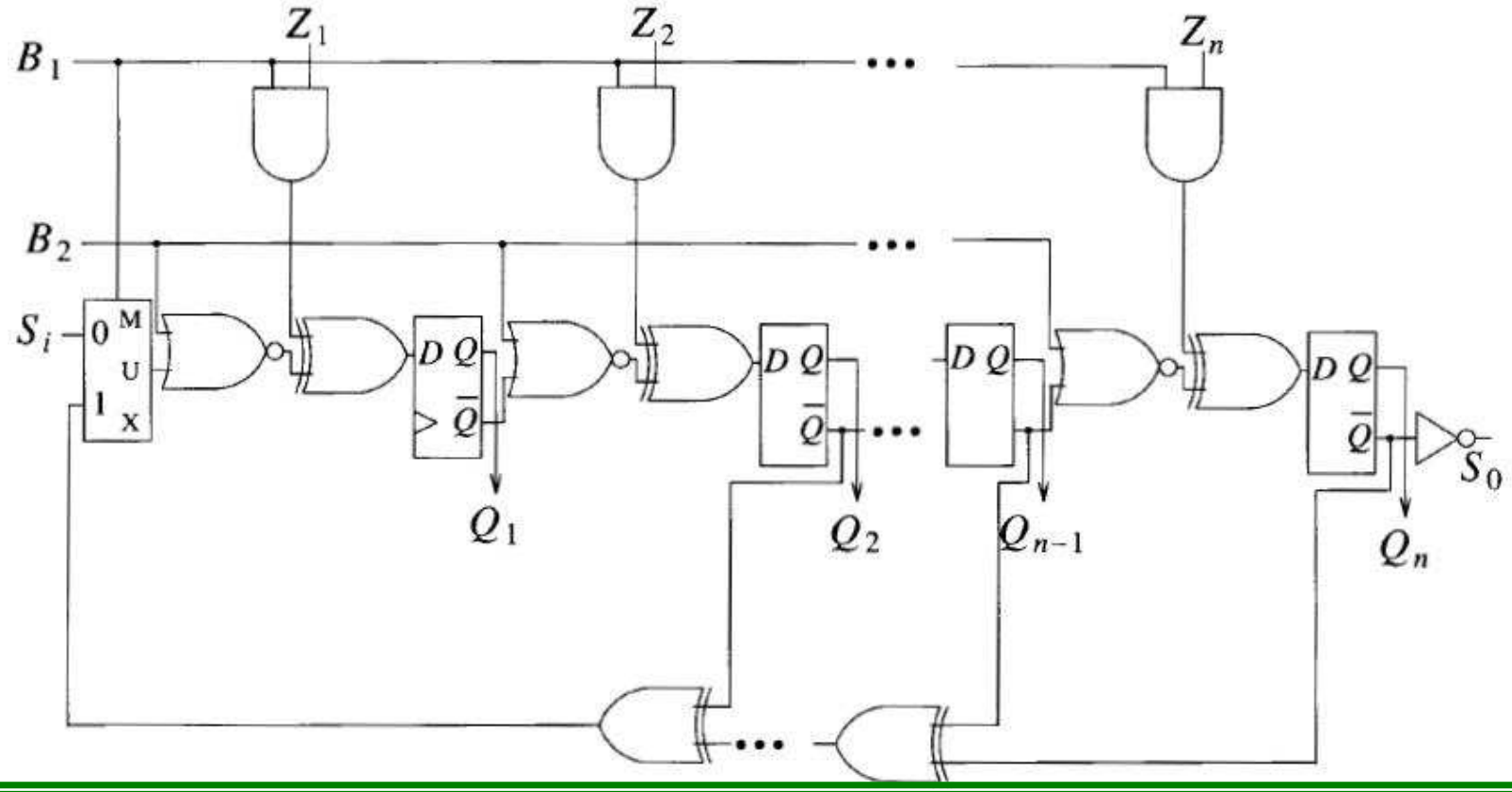
- Test-per-Scan BIST
  - Hardware overhead is low
- Test-per-Clock BIST
  - Execute tests faster than Test-per-Scan BIST
  - More hardware overhead

BILBO

- BILBO stands for built-in logic-block observation.
- In this architecture, the storage cells are clustered into groups known as registers.
- In general these registers are functional registers.
- BILBO takes advantage of the register aspects which gains in more effective test methodology.

BILBO Register

- The BILBO register is shown below:



Contd …

- Z1, Z2, ….Zn → input data
- B1, B2 → control inputs
- Si → Scan input or test data
- Q1, Q2, ….Qn → flip-flop outputs

The input data is loaded parallel depending on the control inputs.

Operation of BILBO

The operation of BILBO is dependent on the control inputs B1, B2.

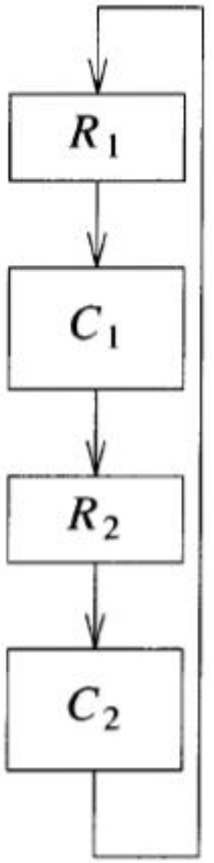
- B1=0, B2=0 → test mode with scan input Si
- B1=0, B2=1 → all storage cells are reset to 0
- B1=1, B2=0 → works as a LFSR
- B1=1, B2=1 → normal mode, data is loaded in parallel through Zi.

BILBO BIST Architecture

- A circuit is partitioned into a set of registers and combinational circuits.
- These registers are replaced by BILBO registers.
- The inputs to combinational circuits are driven by BILBO registers and the outputs drive another BILBO register.

Contd …

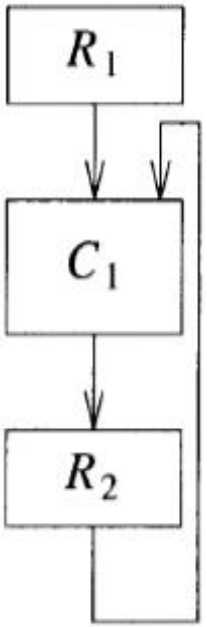
- An architecture is shown in figure.
- The registers are all BILBOs.
- **To test C1**: R1 is put in PRPG mode and R2 is put in MISR mode.
  After the test session, R2 can be scanned out and the signature can be checked.
- **To test C2:** Similarly R2 is put in PRPG mode and R1 in MISR mode.



Thus the circuit is tested in 2 sessions.

Contd …

- Another BILBO BIST architecture is shown in figure.
- **To test C1**: R1 must be in PRPG mode and R2 should be in both MISR and PRPG mode since BILBO R2 have a self loop.

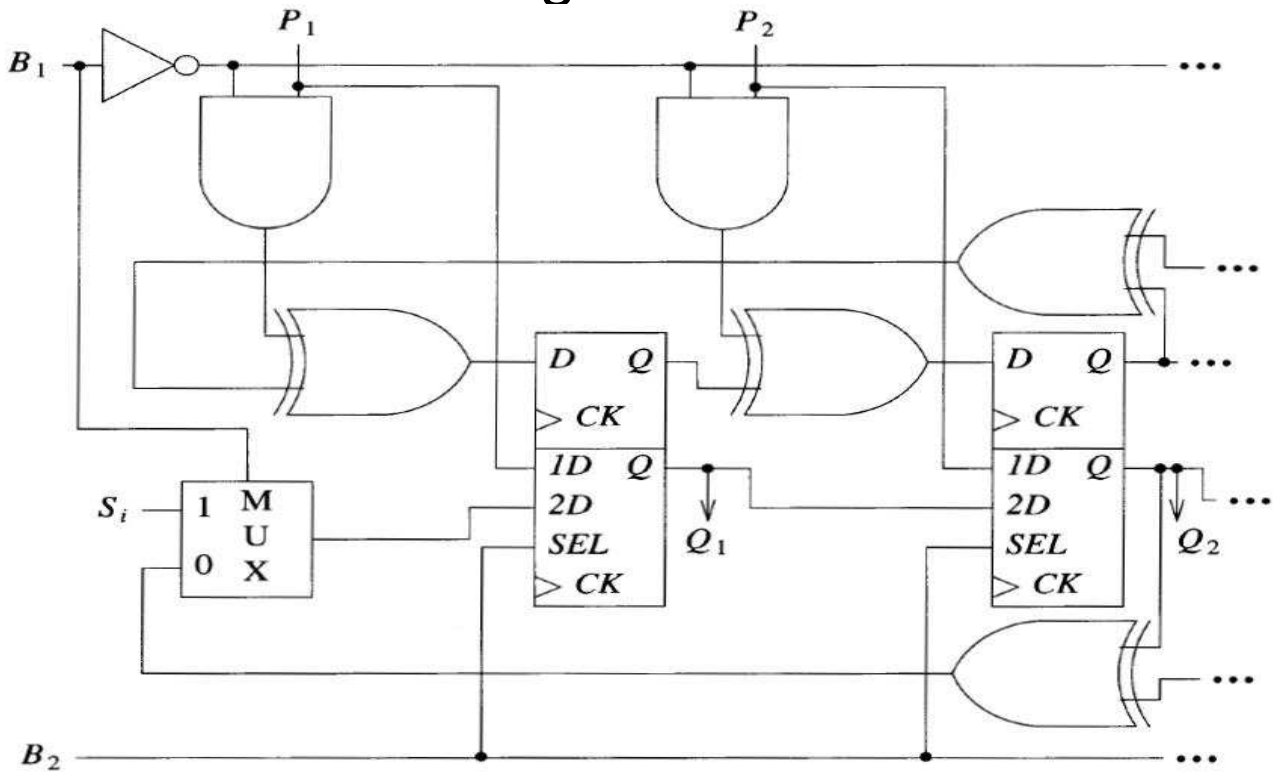


With the BILBO register design given earlier, it cannot be in both MISR and PRPG modes.

To solve this problem concurrent BILBO is developed.

Concurrent BILBO (CBILBO)

- It can be operated simultaneously as PRPG and MISR modes. It is shown in figure below:



Contd …

- The top row of D flip-flop and associated logic form MISR.
- The bottom row of dual-port flip-flops and associated logic form PRPG.
- When B1=0, B2=1 it works in both PRPG and MISR modes.
- When B1=1, B2=1 it works in test mode.

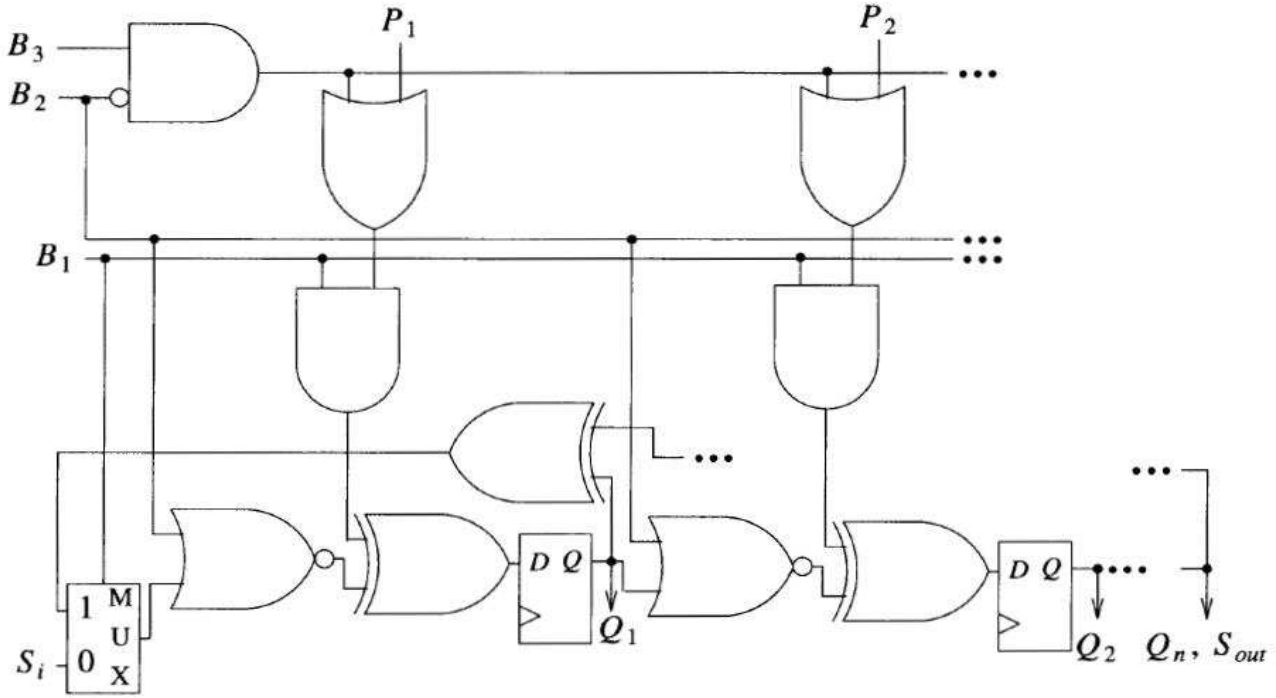
Contd …

- Note that the inputs of a BILBO should be some constant value when it is in PRPG mode.
- Since CBILBO can operate in both PRPG and MISR modes simultaneously, problem arises because the inputs keep changes.
- Hence when BILBO is in PRPG mode, its inputs should be deactivated.

  For this a modified BILBO register with 3 control inputs is designed.

BILBO with 3 control inputs (modified BILBO)

- There will be 8 possible states, so one can be used to activate MISR mode and one can be used for PRPG.

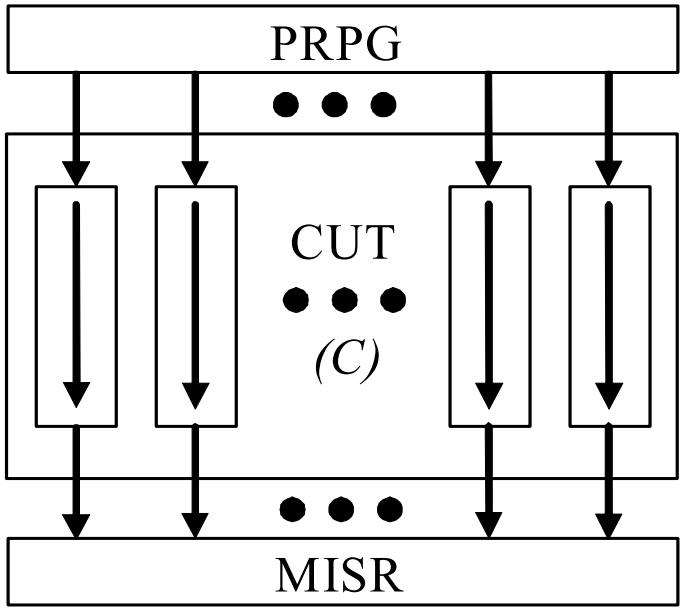


Modes of Operation

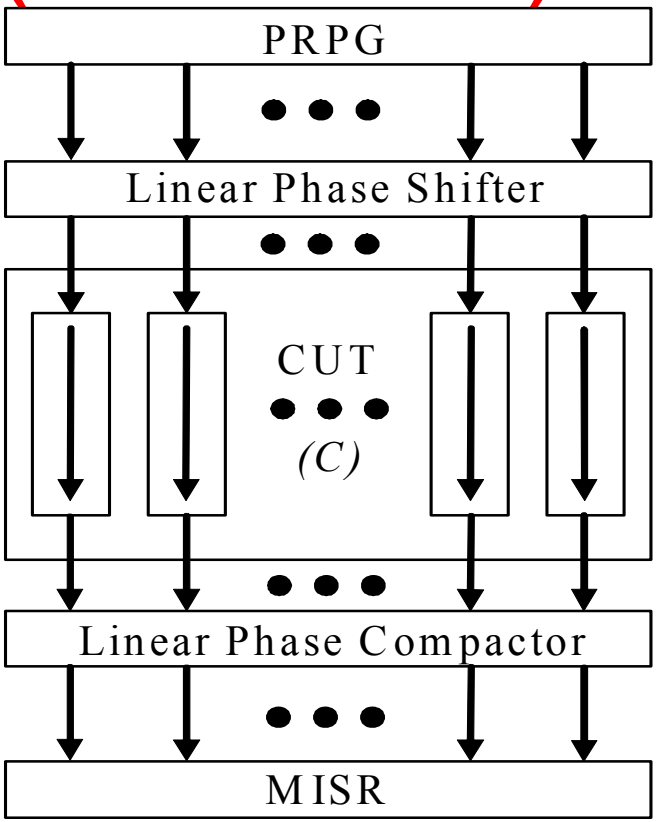
- Since there are 3 control inputs, there will be 8 possible states. Some states are used to specify different modes of BILBO as shown in table.

B1	B2	B3	Mode of Operation
0	0	0	Scan mode
1	0	0	Signature analysis (MISR)
1	0	1	Pattern generation (PRPG)
1	1	0	Normal mode
0	1	0	Reset

Self-Testing Using MISR and Parallel SRSG (STUMPS)

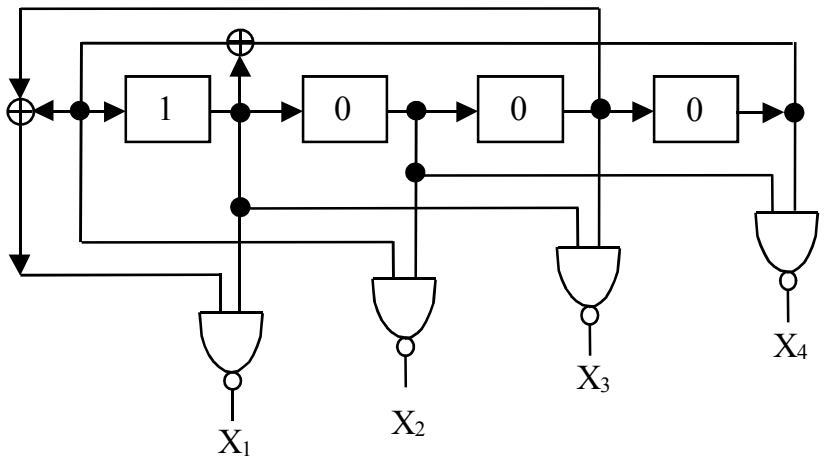


STUMPS



A STUMPS-based architecture

Weighted Pattern Generation



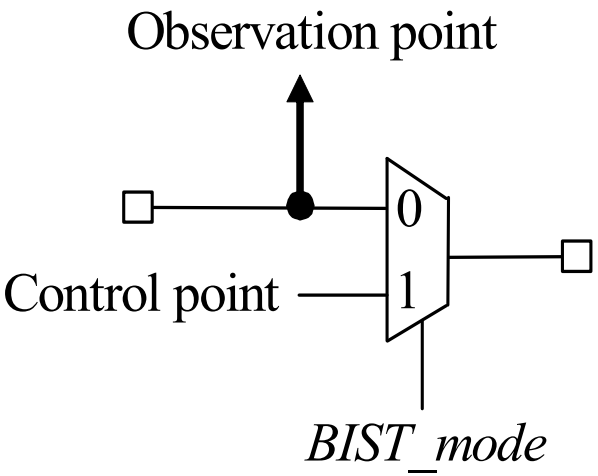
Employ an LFSR

Insert a combinational circuit between the output of LFSR and the CUT

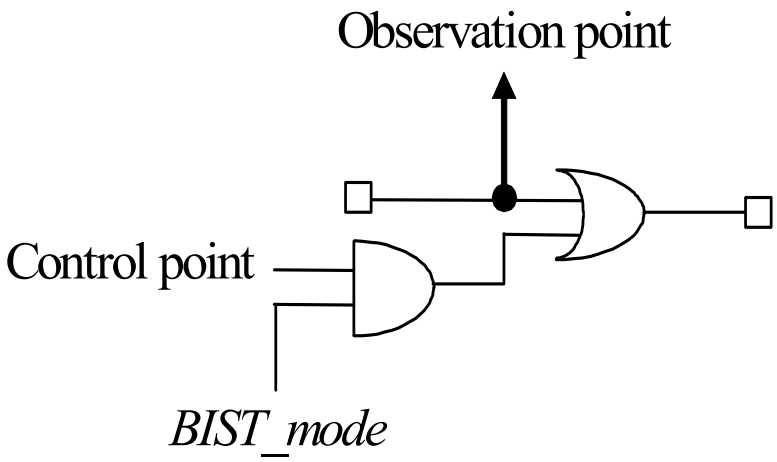
Skew the LFSR probability distribution of 0.5 to either 0.25 or 0.75

Example weighted LFSR as PRPG

Test Point Insertion



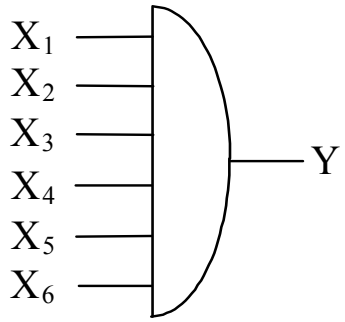
(a) Test point with a multiplexer



(b) Test point with AND-OR gates

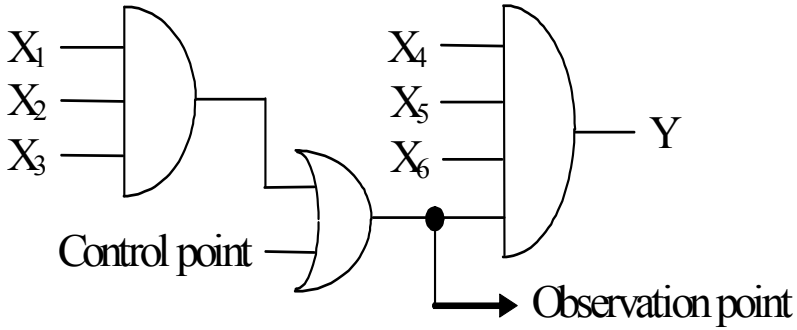
Typical test point inserted for improving a circuit's fault coverage

Example of Inserting Test Points to Improve Detection Probability



$$\text{Min. Detection Probability} = \frac{1}{64}$$

(a) An output RP-resistant stuck-at-0 fault



$$\text{Min. Detection Probability} = \frac{7}{128}$$

(b) Example inserted test points

Test Point Insertion

- Test Point Placement
  - Use fault simulation
  - Use testability measures to guide them
- Control Point Activation
  - During normal operation
    - Deactivated
  - During testing
    - Random activation
    - Deterministic activation

Mixed-Mode BIST

- ROM Compression
  - Store deterministic patterns in ROM
- LFSR Reseeding
  - Generate deterministic patterns by reseeding LFSR with computed seeds
- Embedding Deterministic Patterns
  - Transform the “useless” patterns into deterministic patterns

Hybrid BIST

- Perform top-up ATPG for the faults not detected by BIST
- Store the patterns directly on the tester
- Store the patterns on the tester in a compressed form and make use of the existing BIST hardware to decompress them

Low-Power Logic BIST Architecture

- Low-Transition BIST Design
  - Insert an AND gate and a toggle flip-flop at the scan input of the scan chain
  - Advantages:
    - Less design intrusive
    - no performance degradation
    - Low hardware overhead
  - Disadvantages:
    - Low fault coverage
    - Long test sequence

Low-Power Logic BIST Architecture

- Test-Vector-Inhibiting BIST Design
  - Inhibit the LFSR-generated pseudo-random patterns which do not contribute to fault detection from being applied to the CUT
  - Advantages:
    - Reduce test power
    - No fault coverage loss as the original LFSR
  - Disadvantage:
    - High hardware overhead

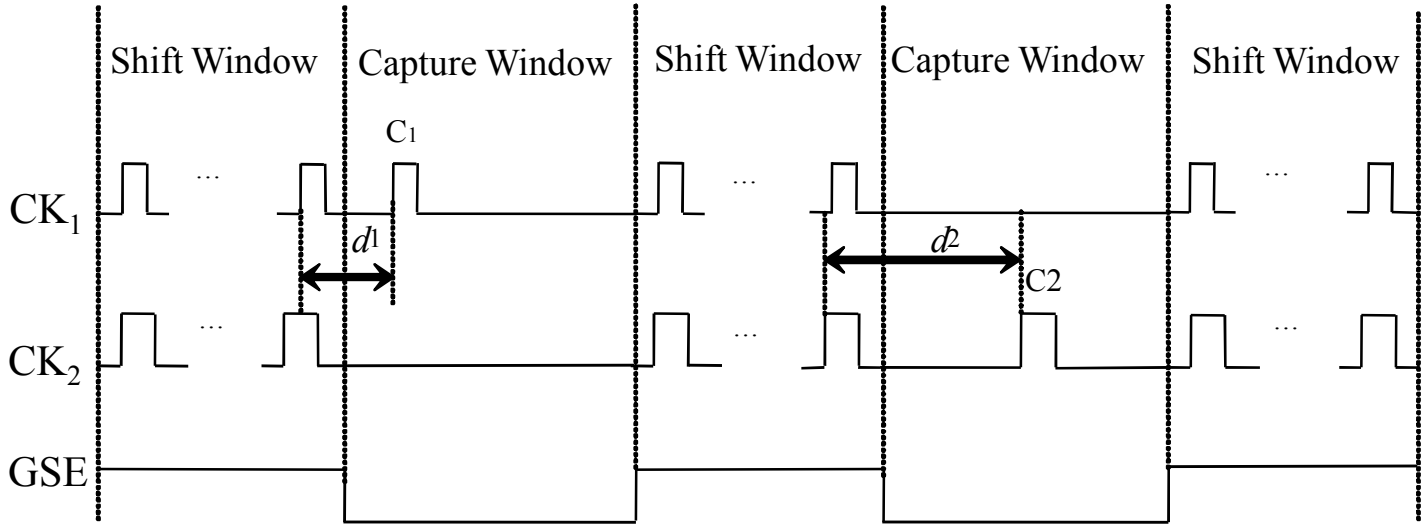
Low-Power Logic BIST Architecture

- Modified LFSR Low-Power BIST Design
  - Use two interleaved n/2-stage LFSRs
  - Advantages:
    - Shorter test length
    - High percentage of power reduction
    - No performance degradation
    - No test time increase
  - Disadvantage:
    - Require constructing special clock trees

At-Speed Logic BIST Architectures

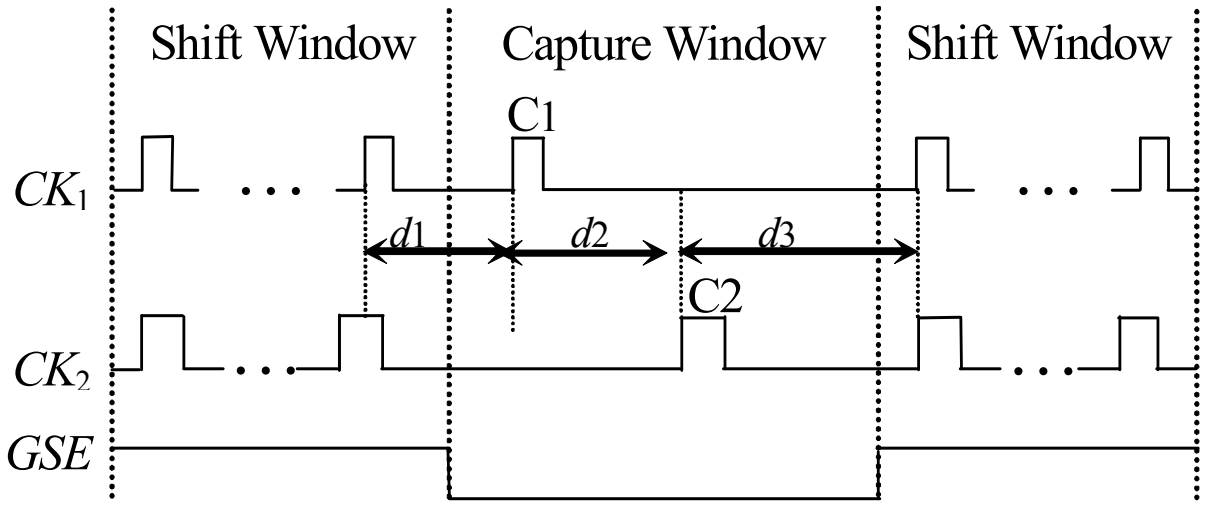
- Single-capture
  - One-hot single-capture
  - Staggered single-capture
- Skewed-load
  - One-hot skewed-load
  - Aligned skewed-load
  - Staggered skewed-load
- Double-capture
  - One-hot double-capture
  - Aligned double-capture
  - Staggered double-capture

One-Hot Single-Capture



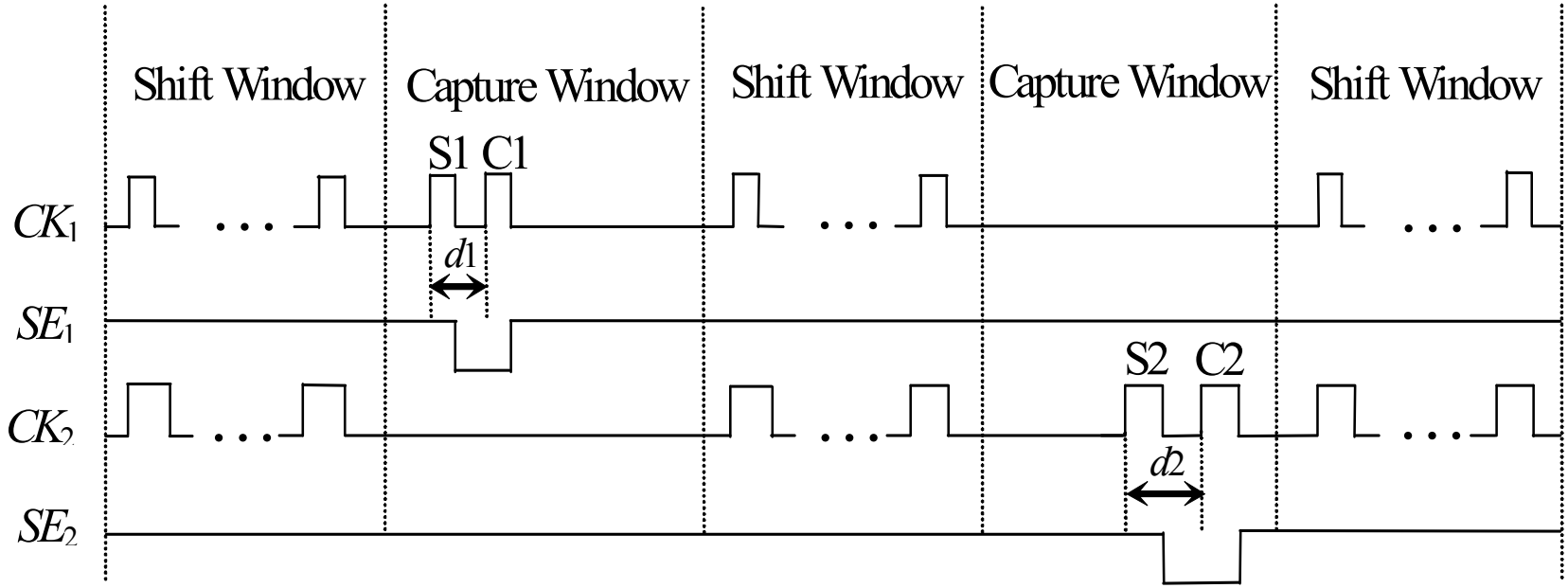
- Advantages:
  - No need to worry about clock skews between clock domains
  - Can be used for slow-speed testing
  - Use a global scan enable (GSE) signal – compatible with Scan
- Disadvantage:
  - Long test time

Staggered Single-Capture



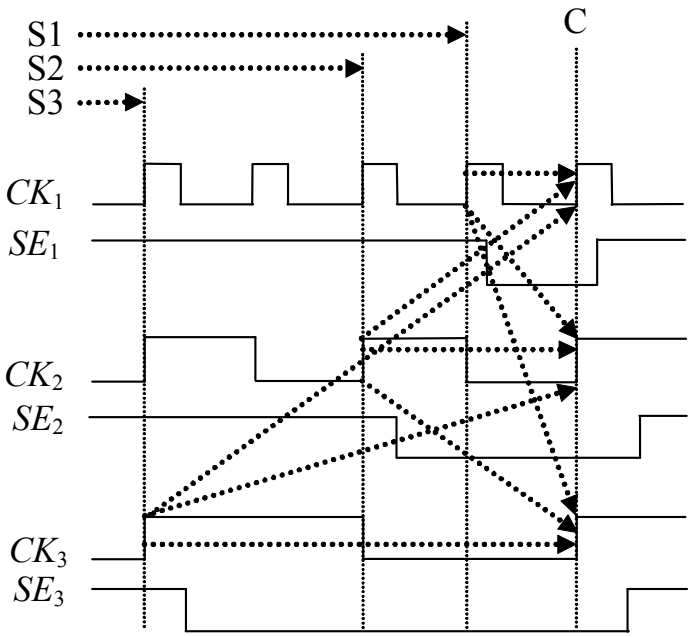
- Advantage:
  - Can detect inter-clock-domain delay faults within two clock domains
- Disadvantage:
  - May cause some structural fault coverage loss if the sequence order of the capture clocks is fixed.

One-Hot Skewed-Load

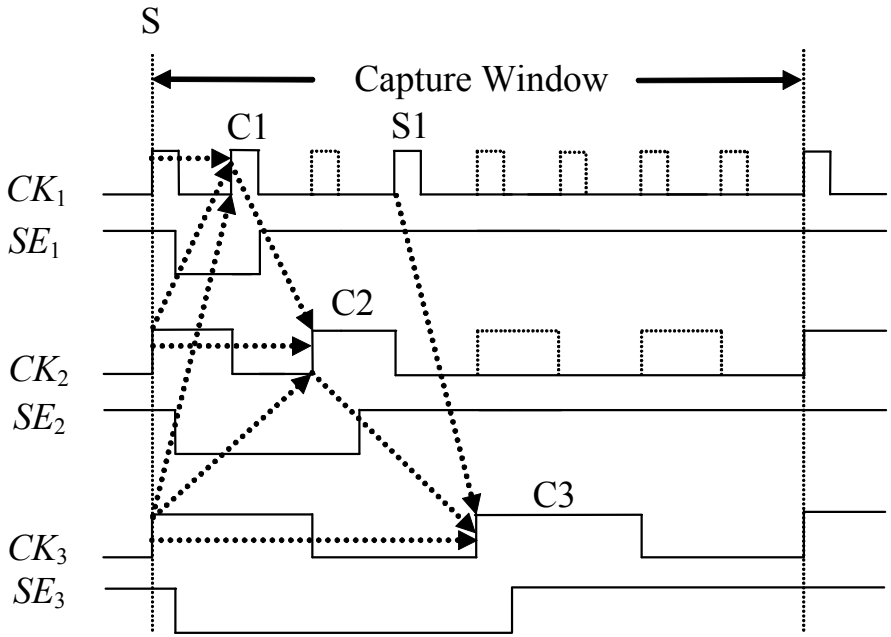


- Advantage:
  - Can be used for at-speed testing of intra-clock-domain delay faults
- Disadvantages:
  - Cannot be used for testing of inter-clock-domain delay faults
  - Long test time

Aligned Skewed-Load



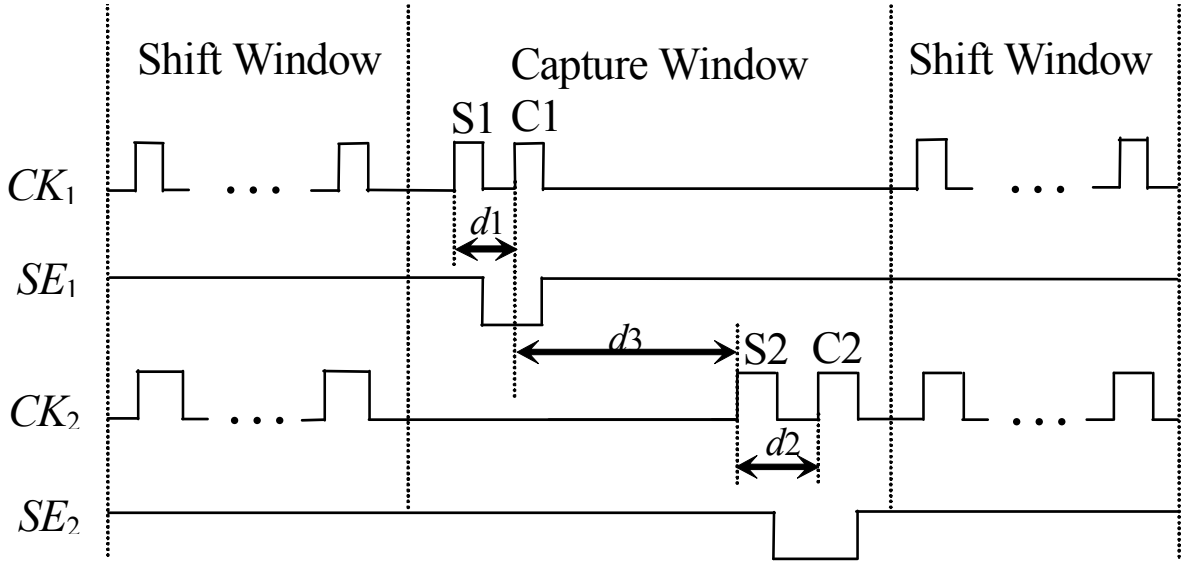
Capture aligned skewed-load



Launch aligned skewed-load

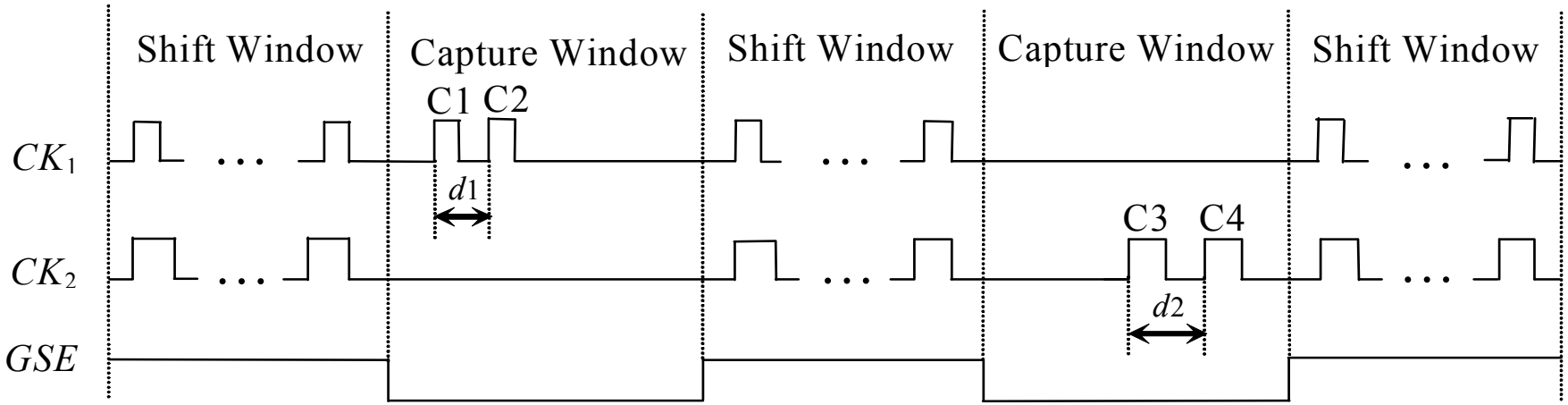
- Advantage:
  - All intra-clock-domain and inter-clock-domain faults can be tested in synchronous clock domains
- Disadvantage:
  - Require more complex timing-control diagram

Staggered Skewed-Load



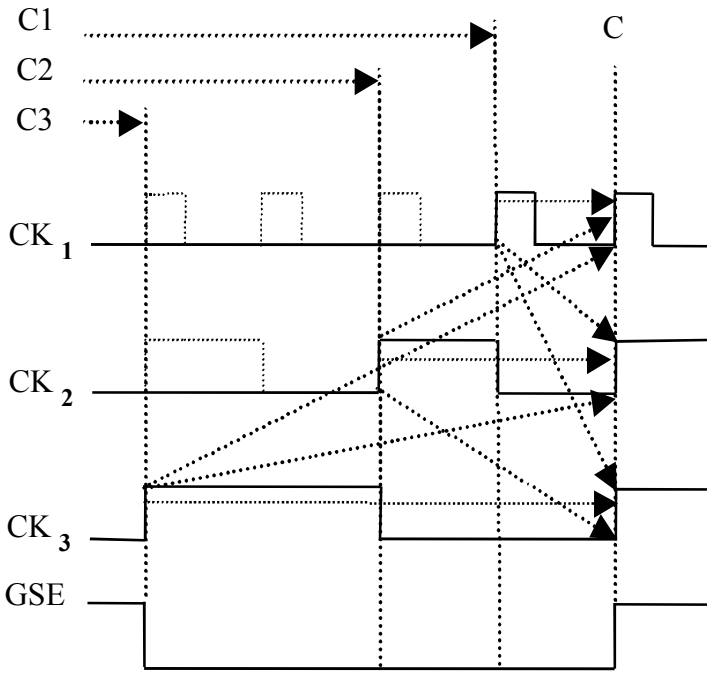
- Advantage:
  - All intra-clock-domain and inter-clock-domain faults can be tested in both synchronous and asynchronous clock domains.
- Disadvantage:
  - Complicated physical implementation

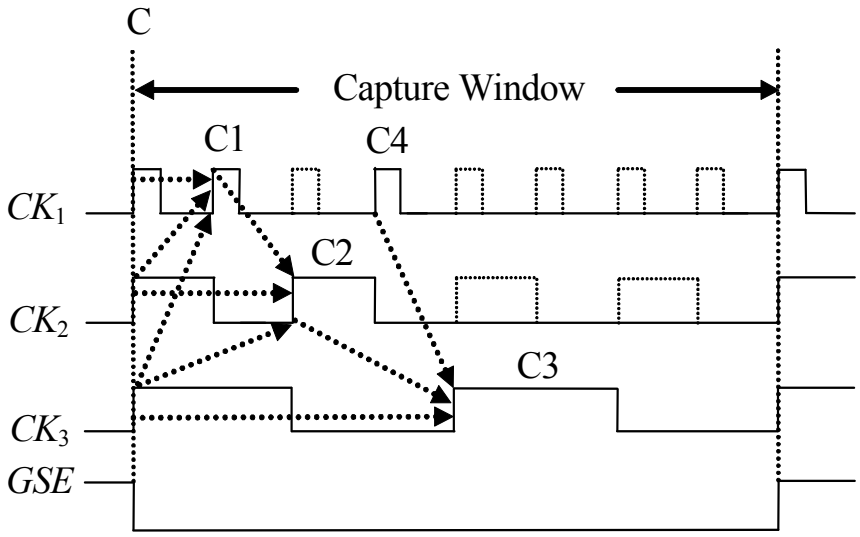
One-Hot Double-Capture



- Advantage:
  - Can be used for true at-speed testing of intra-clock-domain delay faults
- Disadvantages:
  - Cannot be used for testing of inter-clock-domain delay faults
  - Long test time

Aligned Double-Capture

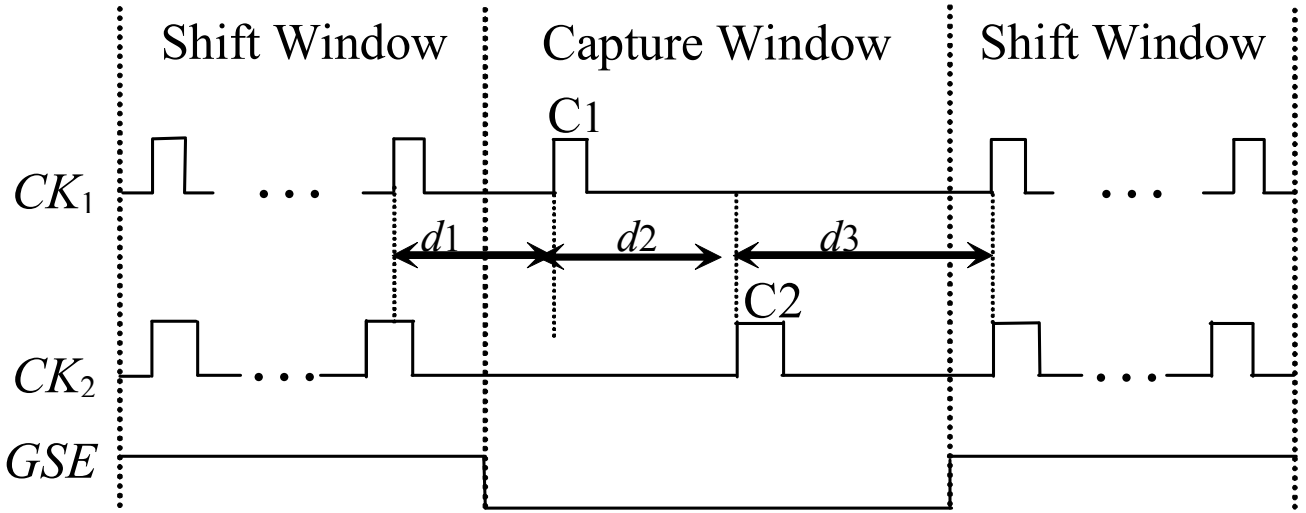




Capture aligned double-capture *Launch aligned double-capture*

- Advantage:
  - Can test all intra-clock-domain and inter-clock-domain delay faults in synchronous clock domains
- Disadvantage:
  - Require precise alignment capture pulses

Staggered Double-Capture



- Advantages:
  - Ease physical implementation
  - Integrate logic BIST with scan/ATPG
- Disadvantage:
  - May cause fault coverage loss due to the ordered sequence of capture clocks.

Summary of Industry Practices for At-Speed Logic BIST

Industry Practices	Skewed-load	Double-Capture
Encounter Test	Through service	Through service
ETLogic	√	Through service
LBIST Architect	√	√
TurboBIST-Logic		√

Questions?