Lecture and Hands-on Workshop on Design for Test (DFT)

Module 1
Fundamentals of Design for Testability (DFT)
Session 2, part 1: DFT Concepts

Carlos Andrey Morales-Zamora, Sergio Arriola-Valverde, Renato Rimolo-Donadio

Electronics Engineering Department, Instituto Tecnológico de Costa Rica

Sponsored by Intel, Costa Rica

October, 2025

TEC | Tecnológico de Costa Rica

intel.

General Lecture and Workshop Content

Module 1	Module 2	Module 3
Semiconductor fabrication and assembly of integrated circuits	High volume manufacturing test approaches	Hands-on labs on testing
Testing of integrated circuits	Structural and automated test approaches	DFT CAD Tools
Automated test equipment	Automated test pattern generation and standardization	FPGA-based DFT implementation
Introduction to high volume manufacturing flow	Functional content	

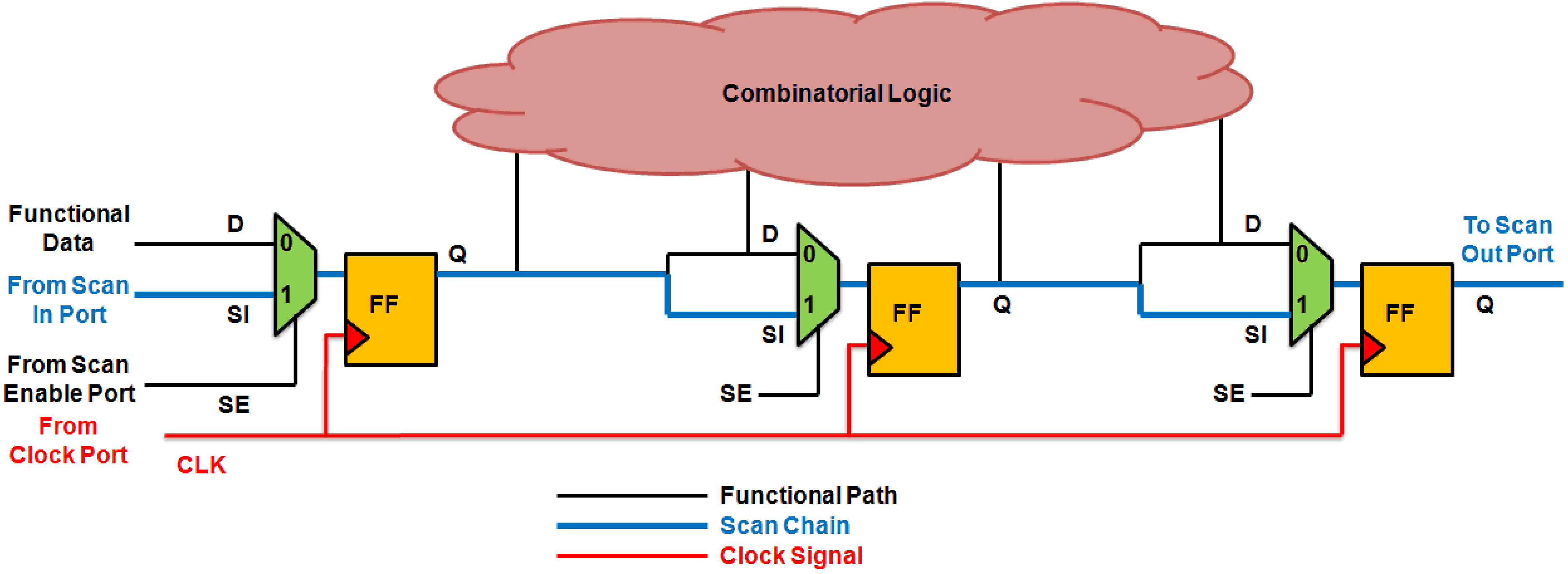
General Lecture and Workshop Content

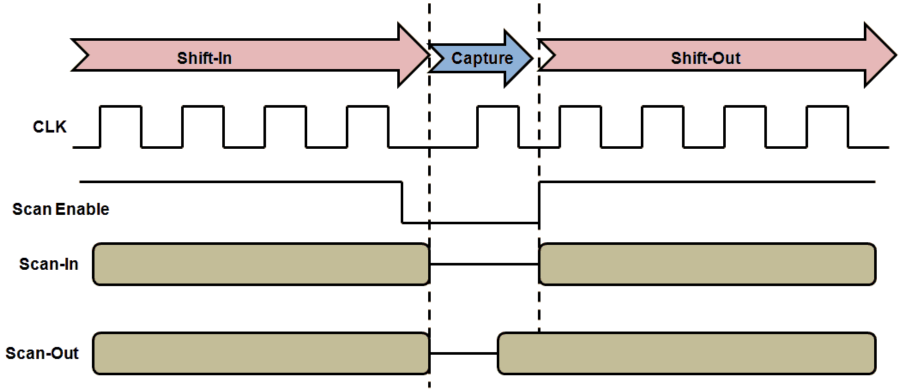
DFT Concepts:

- SCAN
- ATPG
- BIST

Scan Architecture

Method to detect manufacturing faults:

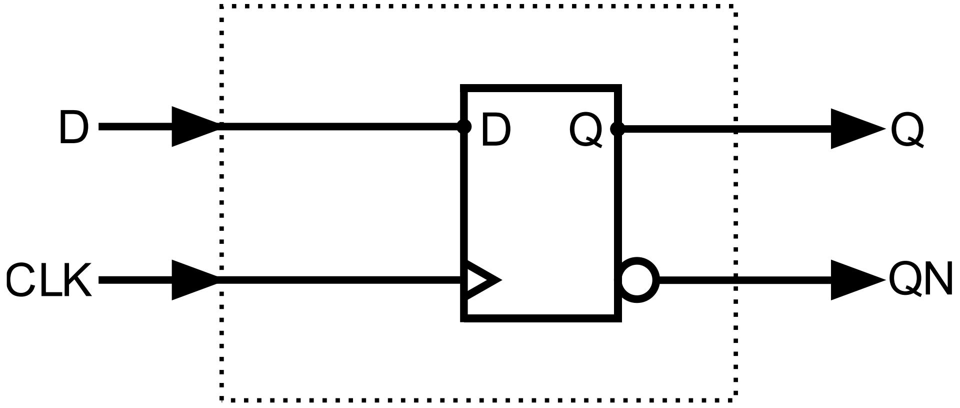




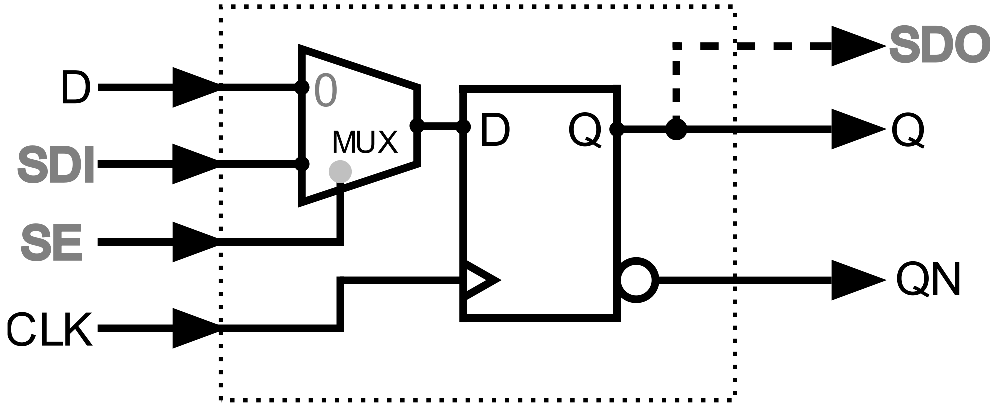
Source: https://anysilicon.com/overview-and-dynamics-of-scan-testing/

The Mux-D Flip-Flops Scan Cell

- This is the most **versatile scan cell**.
- It consists of a Flip-Flops with a **multiplexer** at input D.
- Additional ports:
    - **SDI** (Scan Data Input).
    - **SE** (Scan Enable).
    - **SDO** (Scan Data Output, can be dedicated or shared with Q/QN).

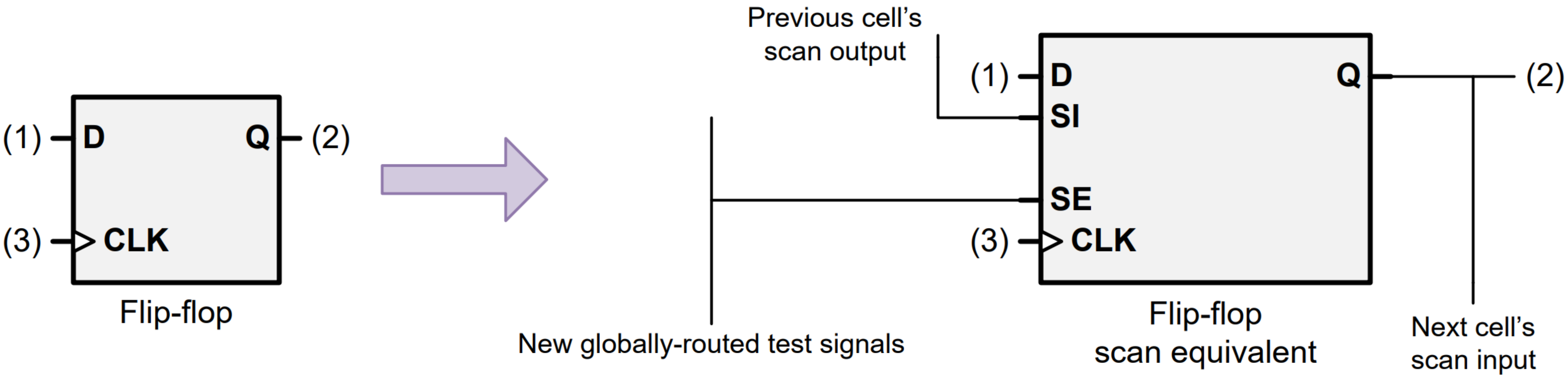


Regular D Flip-Flop



Scannable D Flip-Flop

The Mux-D Flip-Flops Scan Cell



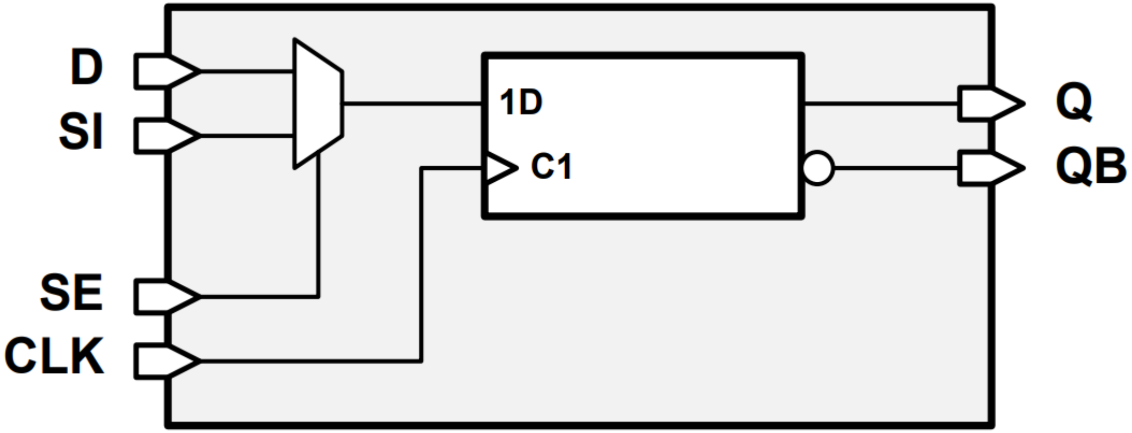
Flip-flop

Previous cell's scan output

New globally-routed test signals

Flip-flop scan equivalent

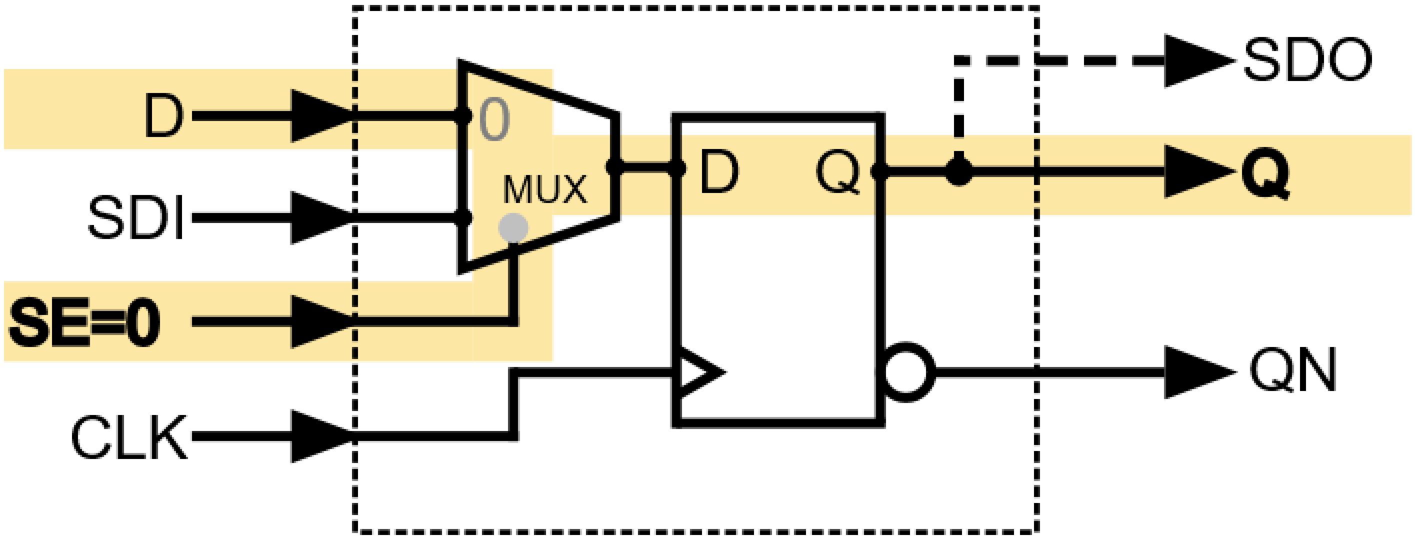
Next cell's scan input



Scan Elements Operations

The scan cell provides observability and controllability of the signal path by conducting the four transfer functions of a scan element.

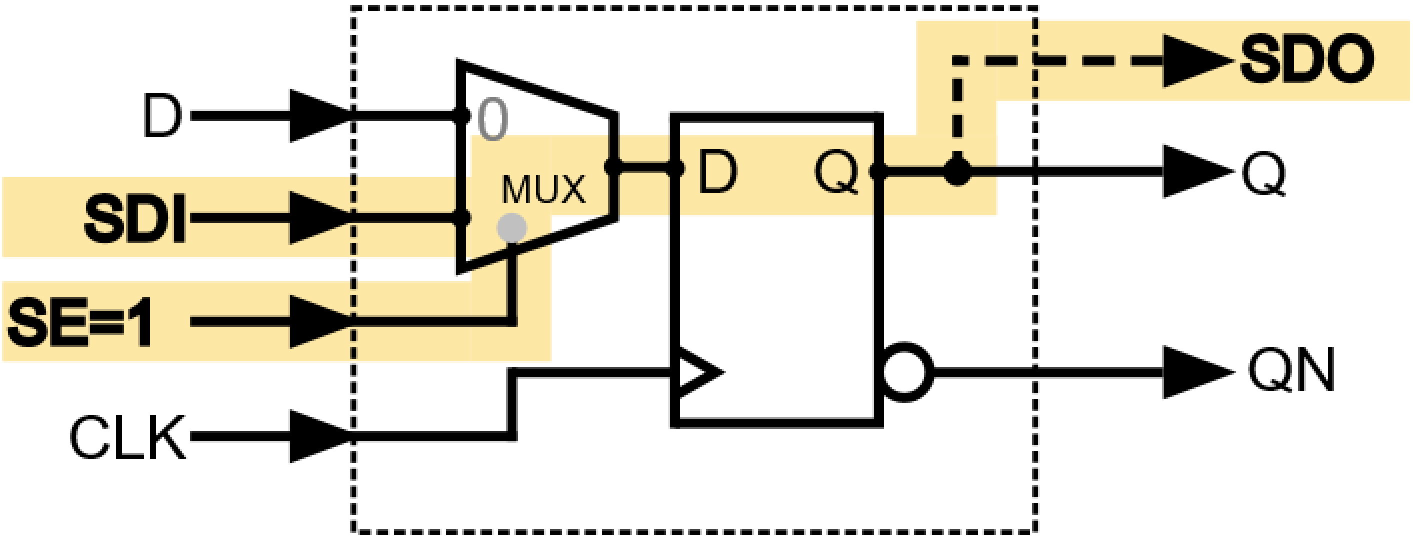
- **Functional mode**: D to Q through port a of the input multiplexer.
  - Allows normal transparent operation of the element.



Scan Elements Operations

The scan cell provides observability and controllability of the signal path by conducting the four transfer functions of a scan element.

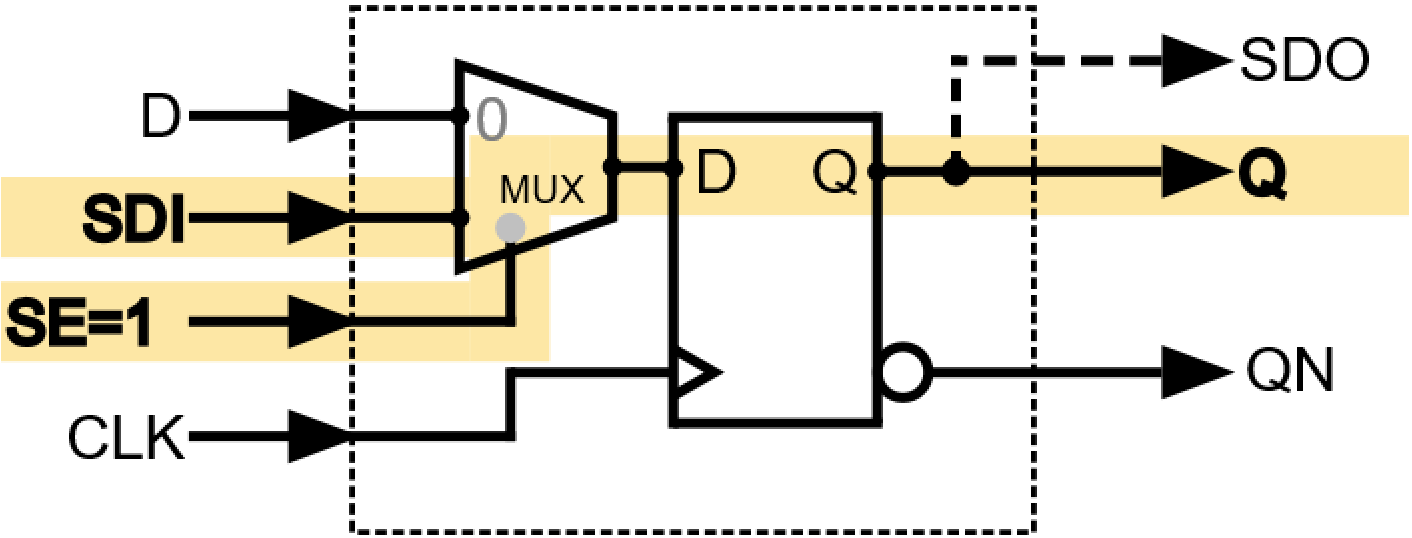
- **Scan Sample**: D to SDO through port a of the input multiplexer.
  - Gives observability of logic that fans into the scan element.



Scan Elements Operations

The scan cell provides observability and controllability of the signal path by conducting the four transfer functions of a scan element.

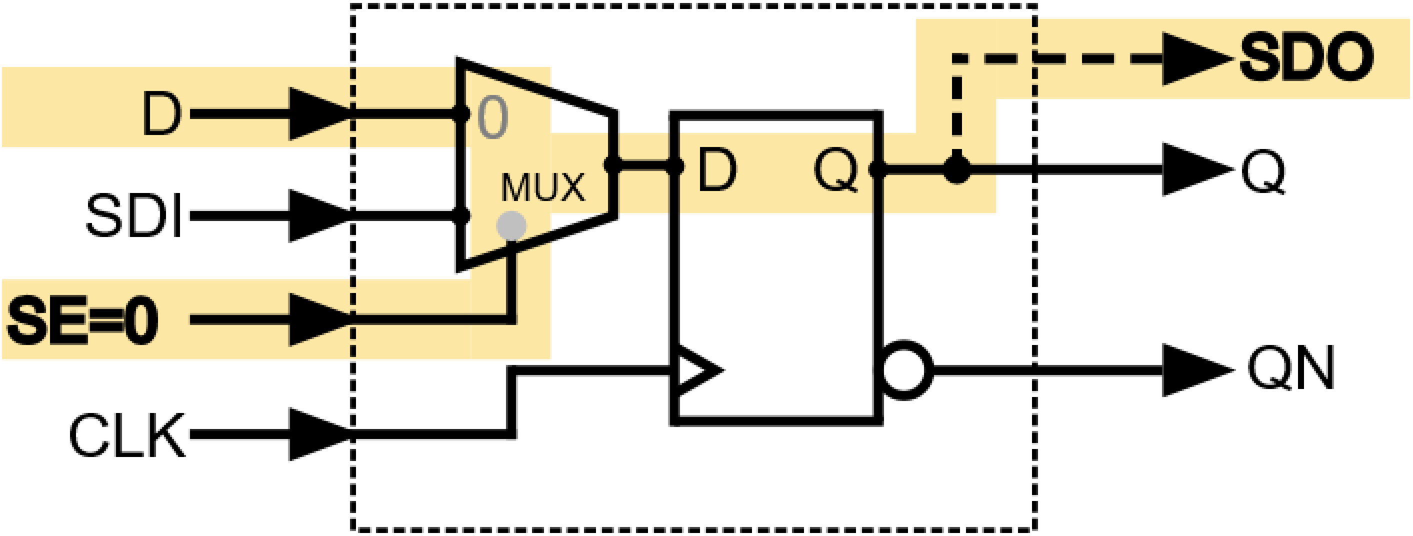
- **Scan Load/Shift**: SDI to SDO through the b port of the multiplexer.
  - Used to serially load/shift data into the scan chain while simultaneously unloading the last sample.



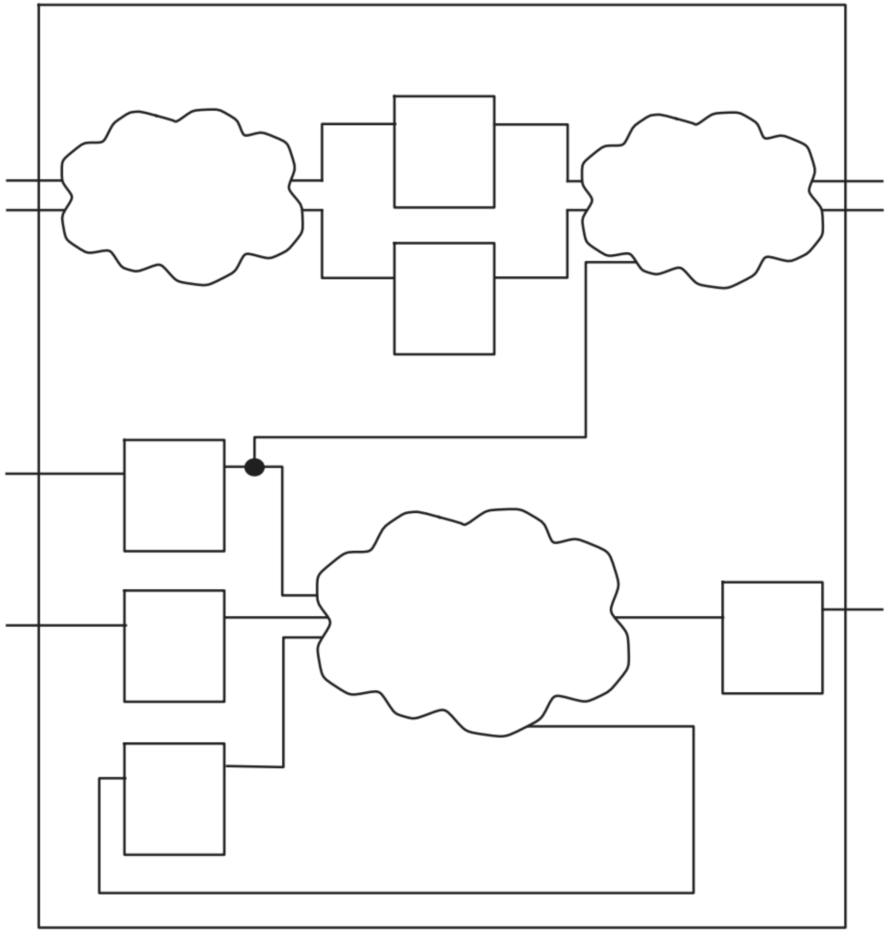
Scan Elements Operations

The scan cell provides observability and controllability of the signal path by conducting the four transfer functions of a scan element.

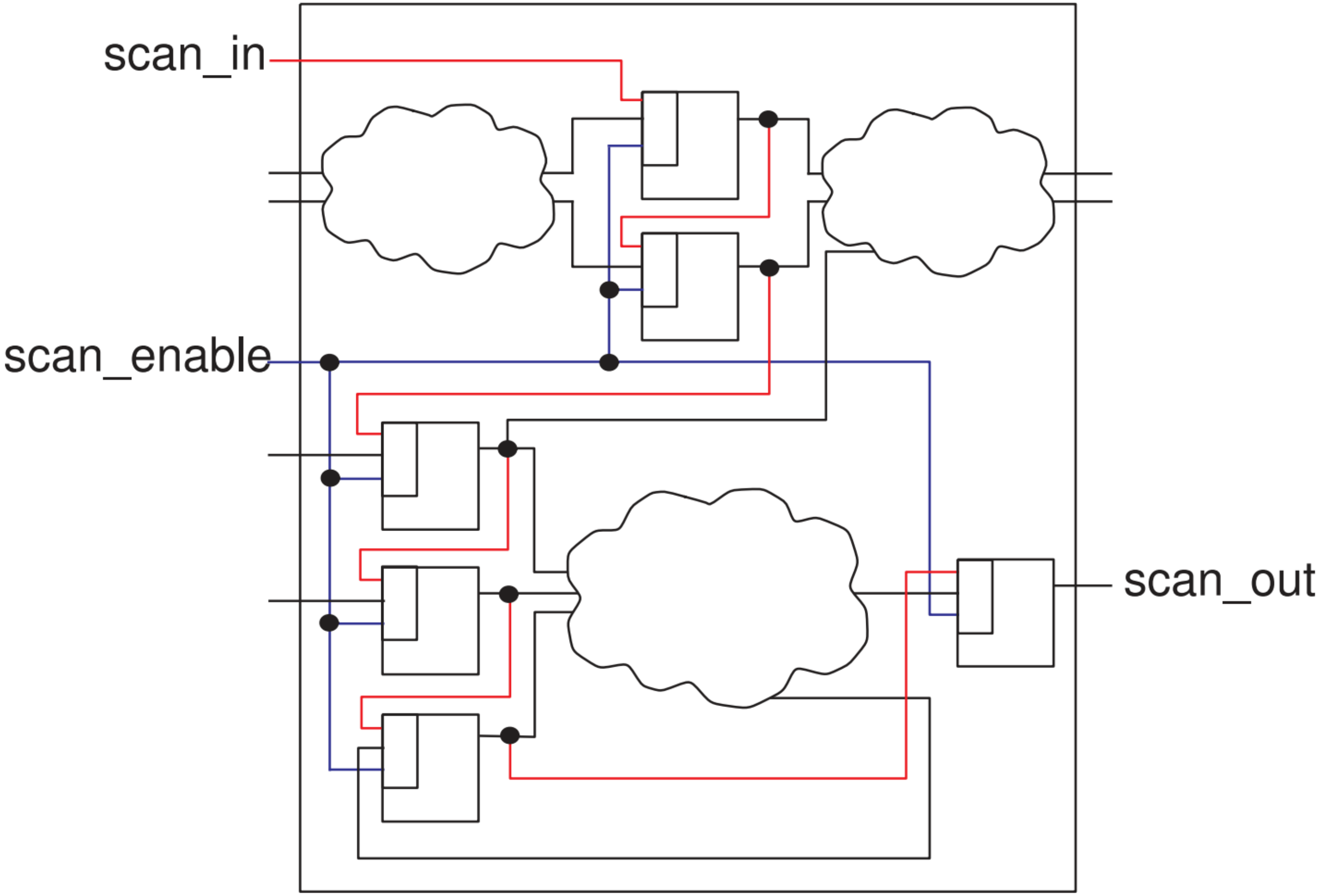
- **Scan Data Apply**: SDI to Q through the b port of the multiplexer:
  - Allows the scan element to control the value of the output, thereby controlling the logic driven by Q.



Full Scan

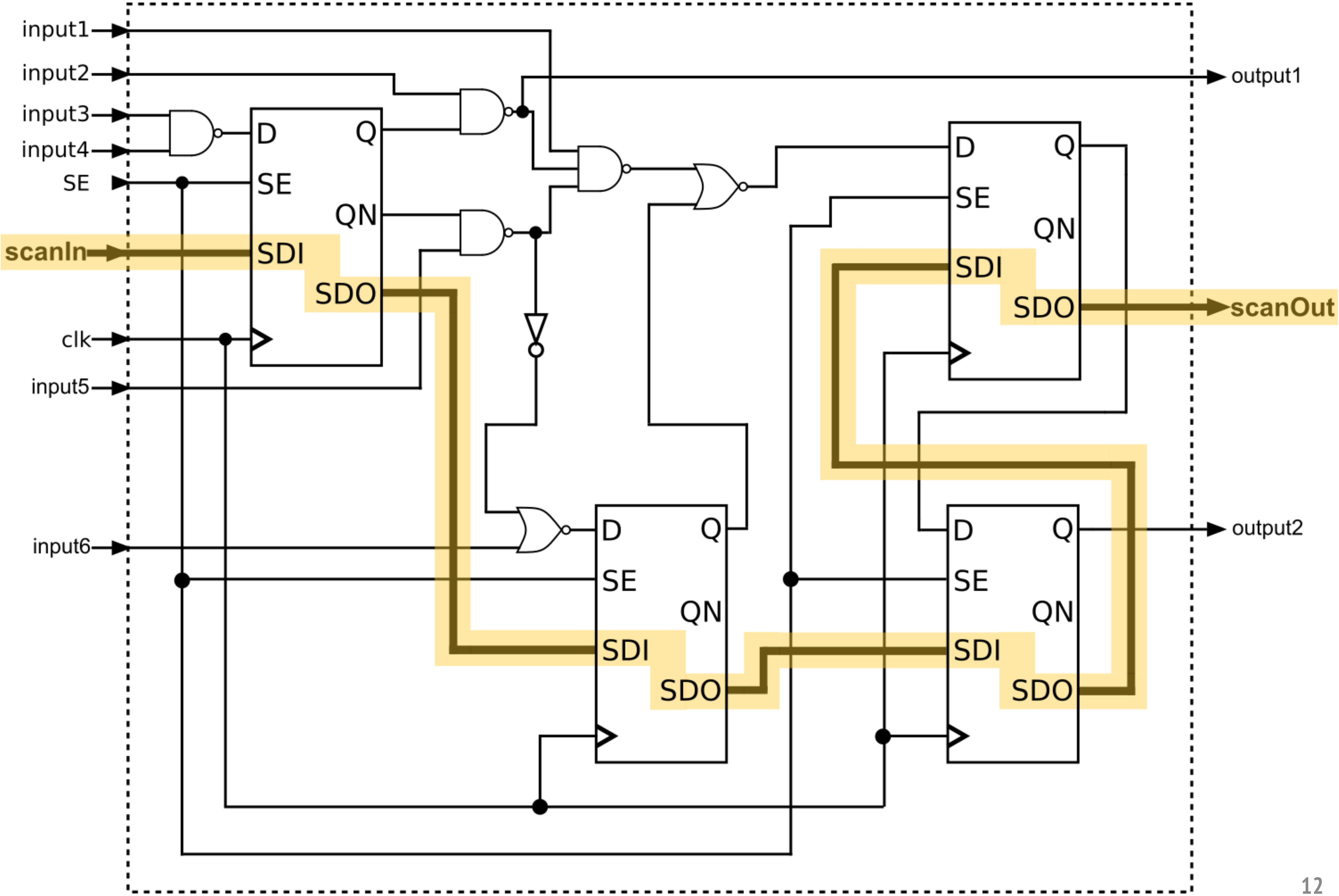


Without Scan



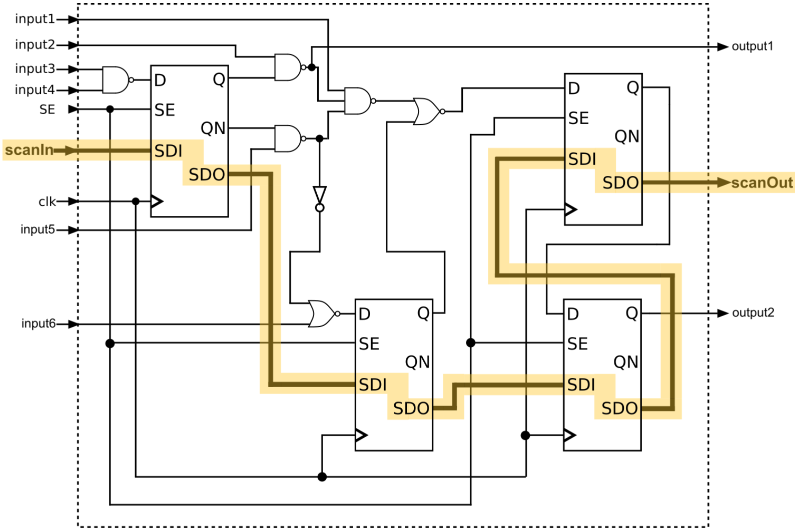
With Full Scan

Full Scan



input1
input2
input3
input4
SE
scanIn
clk
input5
input6
output1
scanOut
output2
D
Q
SE
QN
SDI
SDO

Example in Logisim-evolution



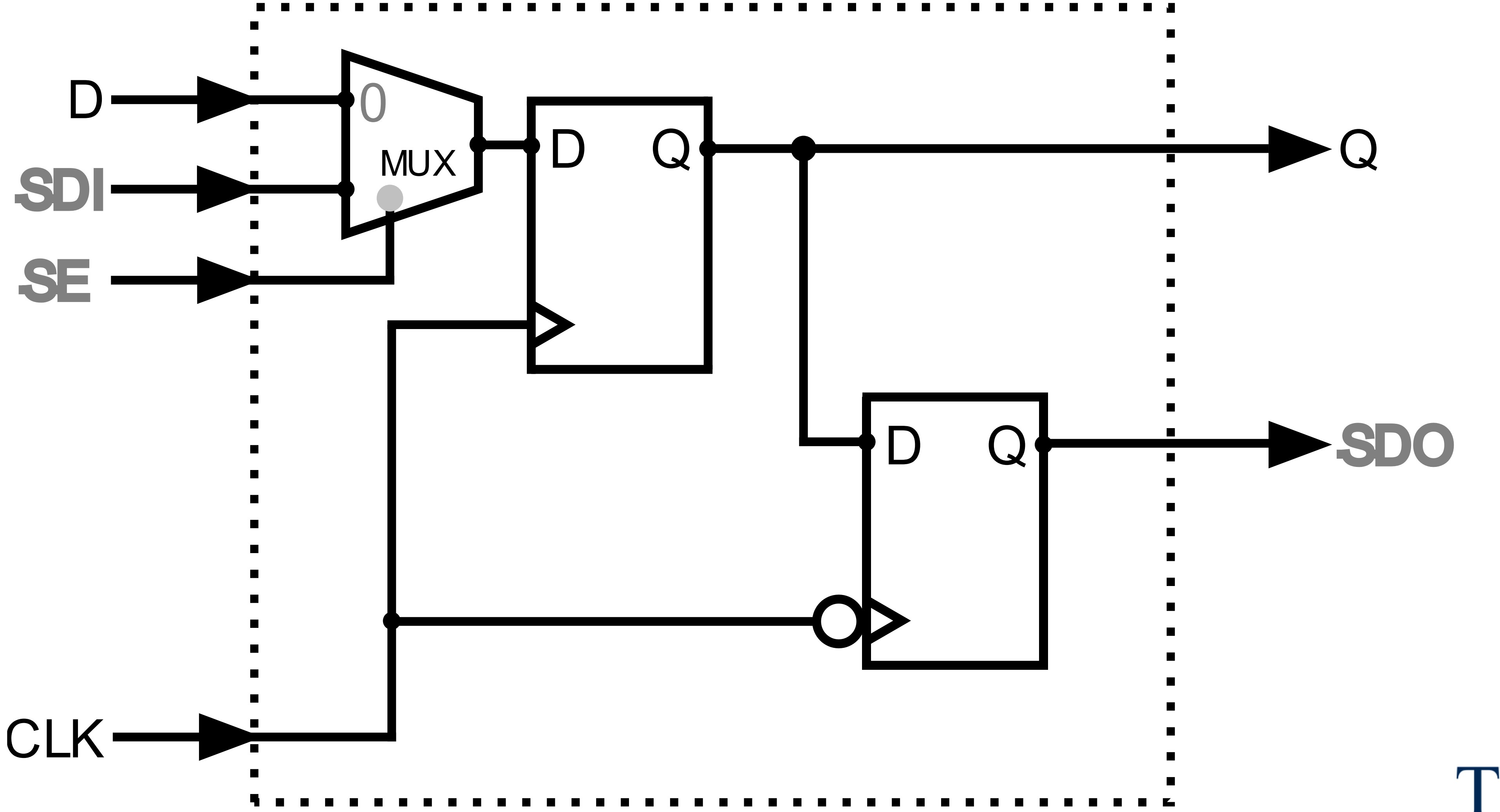
Download link:

https://github.com/logisim-evolution/logisim-evolution

Example:

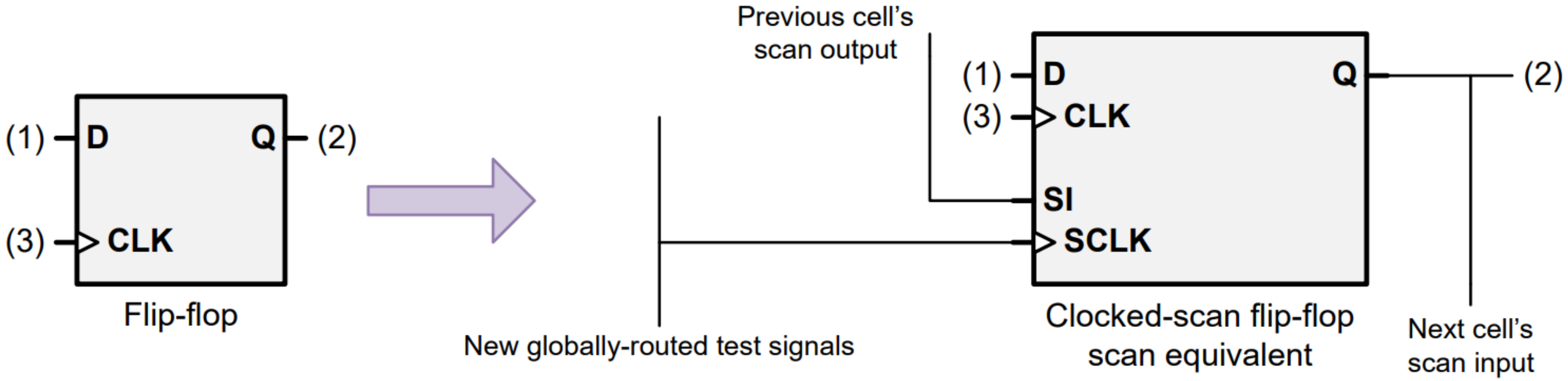
Scan_Example.circ

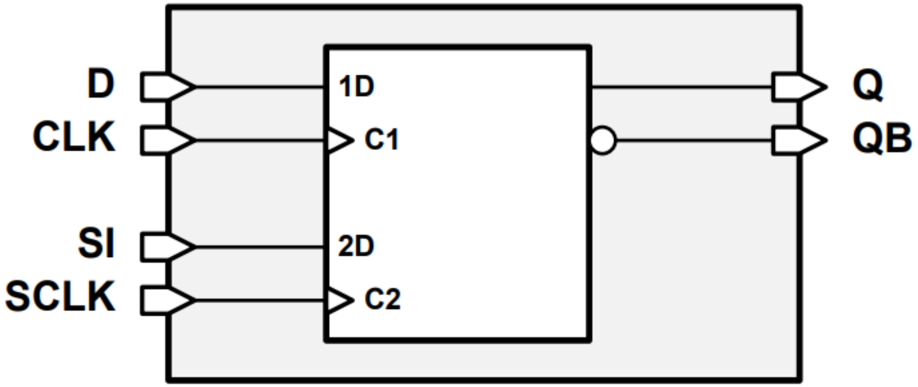
Edge-triggered Scan



Clocked-Scan Scan

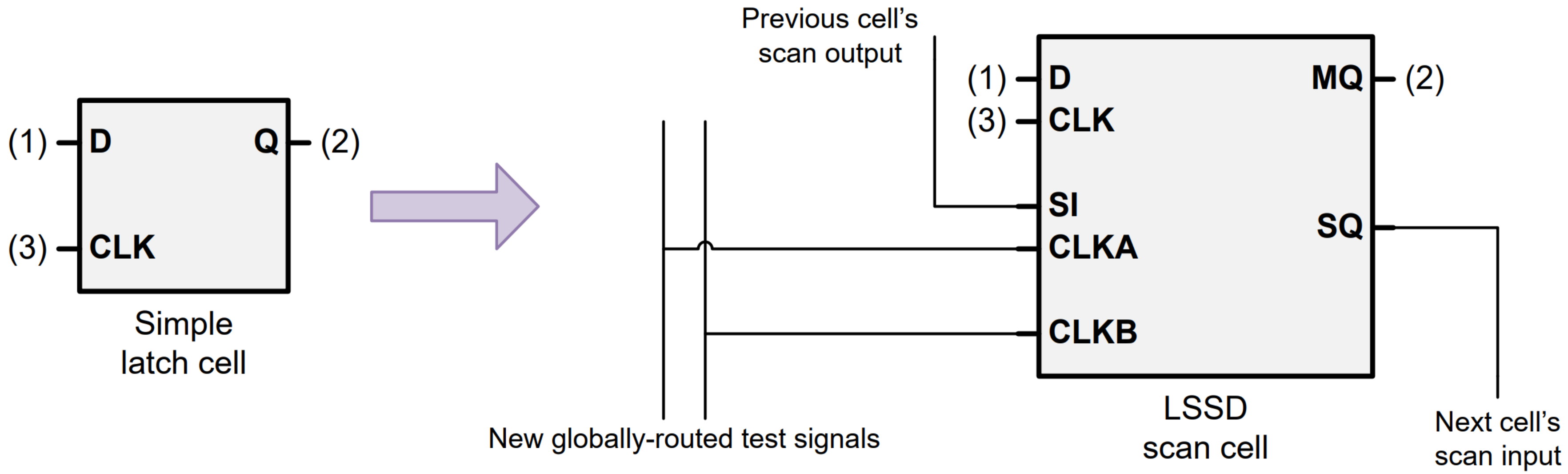
- The input selection is performed by two independent clocks.
- **Functional mode:** the system clock is active and system data is clocked into the cell.
- **Scan shift:** the test clock is active and scan data is clocked into the cell.

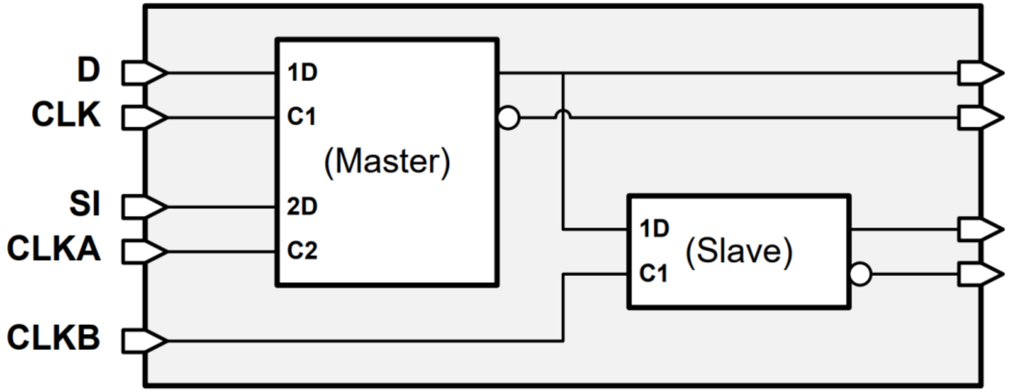

(1)
D
Q
(2)
(3)
CLK
Flip-flop
Previous cell's scan output
(1)
D
Q
(2)
(3)
CLK
SI
SCLK
New globally-routed test signals
Clocked-scan flip-flop scan equivalent
Next cell's scan input



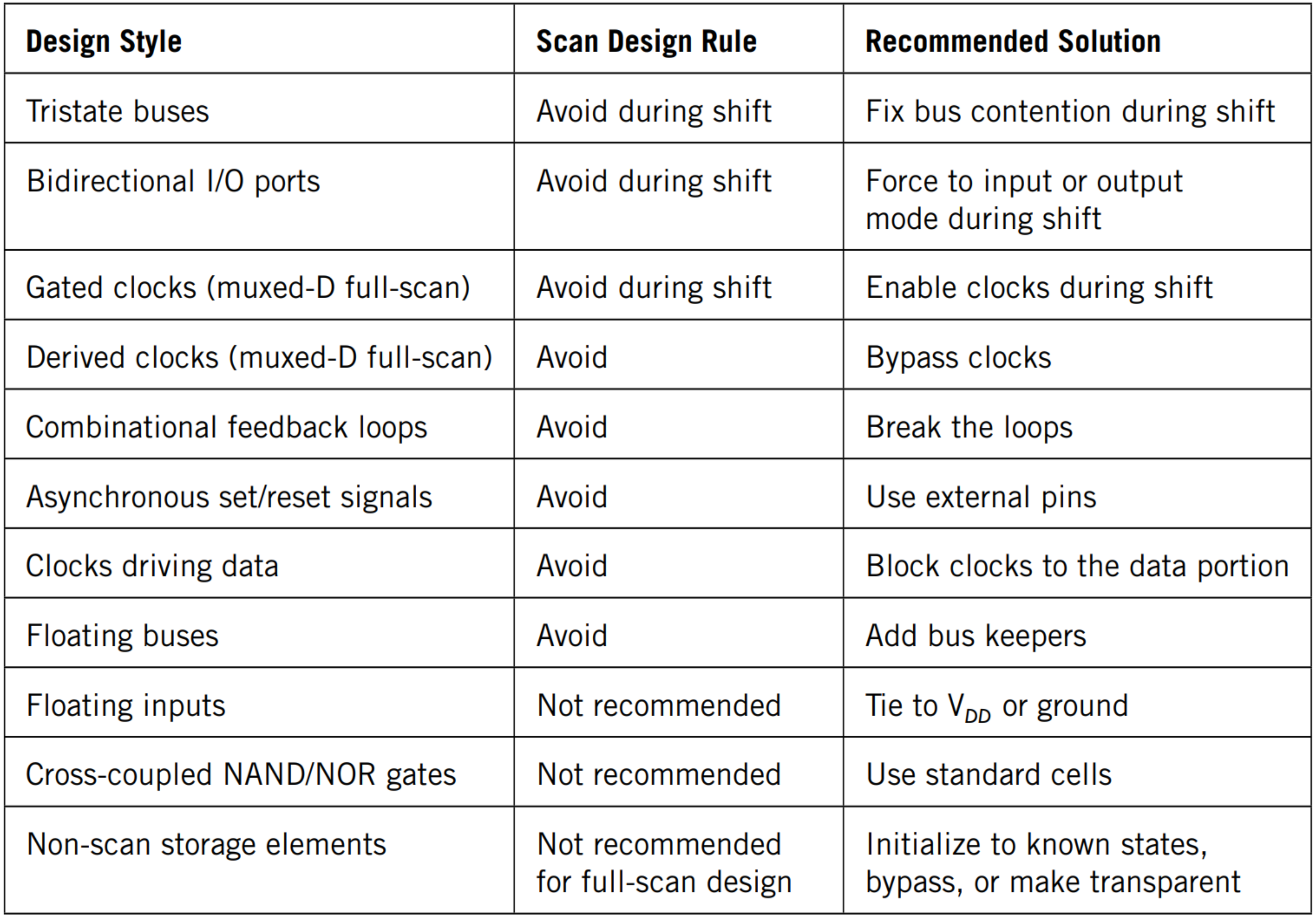
Level-Sensitive Scan Design (LSSD)

- Uses level-sensitive **latches** (instead of edge-triggered flip-flops)
- Data is captured only when the corresponding clock is **active** (It does not depend on the **clock edge**).
- It avoids the **race condition**.



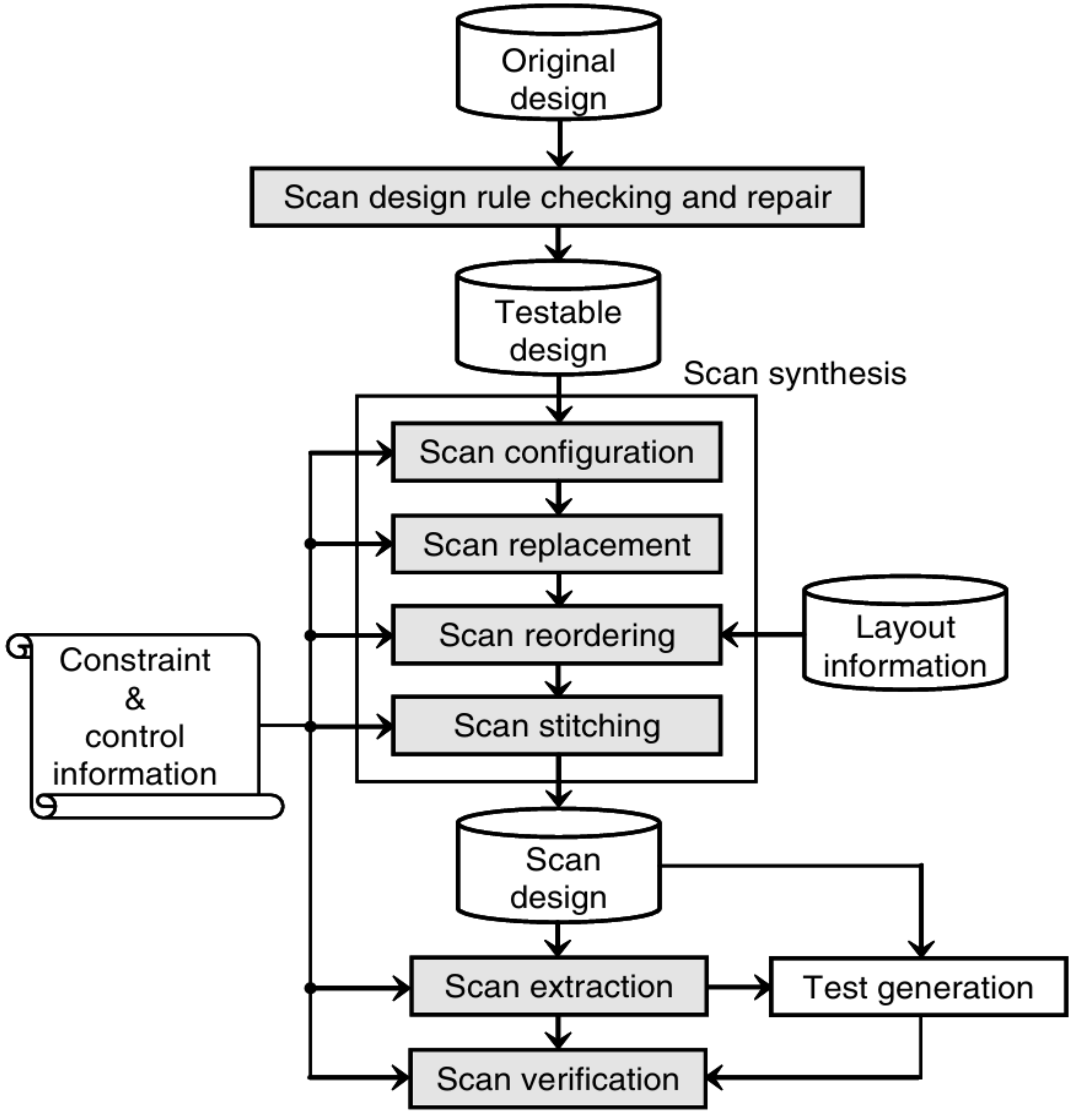


Scan Design Rules



Design Style	Scan Design Rule	Recommended Solution
Tristate buses	Avoid during shift	Fix bus contention during shift
Bidirectional I/O ports	Avoid during shift	Force to input or output mode during shift
Gated clocks (muxed-D full-scan)	Avoid during shift	Enable clocks during shift
Derived clocks (muxed-D full-scan)	Avoid	Bypass clocks
Combinational feedback loops	Avoid	Break the loops
Asynchronous set/reset signals	Avoid	Use external pins
Clocks driving data	Avoid	Block clocks to the data portion
Floating buses	Avoid	Add bus keepers
Floating inputs	Not recommended	Tie to V_{DD} or ground
Cross-coupled NAND/NOR gates	Not recommended	Use standard cells
Non-scan storage elements	Not recommended for full-scan design	Initialize to known states, bypass, or make transparent

Scan Design Flow



A. L. Crouch, "Design-for-Test for Digital IC's and Embedded Core Systems" Upper Saddle River, USA: Prentice Hall, 1999.

Partial-Scan & Multiple Scan Chains

Partial-Scan Challenges

Key Issue:

- Non-scanned elements: Loss of controllability (X states) and observability (unseen captures).
- Reduces fault coverage & increases ATPG runtime/vector count.

Compromise:

- Partial-scan (only critical elements excluded, e.g., timing/area-sensitive).

Multiple Scan Chains

- Instead of a single long exploration chain, the design is divided into several shorter chains that run in parallel.
- This speeds up testing without affecting fault coverage.

Example:

- 1 chain with 1000 flip-flops → 1000 cycles.
- 10 chains with 100 flip-flops each → only 100 cycles.

At-Speed Scan Testing & DC Scan Testing

Aspect	At-Speed Scan Testing	DC Scan Testing
Goal	Detects delay and timing defects.	Detects logic or structural faults.
Speed	Runs at real operating frequency.	Runs at slow (non-functional) speed.
Timing Requirements	Strict – control signals must switch within one clock cycle.	Relaxed – no timing constraints.
Complexity	Higher – needs precise timing and special control signals.	Lower – easier insertion and ATPG generation.
Design Impact	Requires strict scan routing and timing rules.	Fewer restrictions on scan chain design.
ATE Compatibility	Not all testers support high-speed switching.	Works with any tester.
Cost & Complexity	Higher – needs precise timing and validation.	Lower – simpler and cheaper to implement.

Key Takeaways

- At-Speed Scan: tests the chip at its actual speed, **detects delay defects**.
- DC Scan: tests without frequency limits, **detects logic faults**.

Outline

- Testing
- Introduction to DFT
- Scan
- ATPG
- BIST
- IEEE Standards for Embedded Core Testing

Automated Test Pattern Generation (ATPG)

What is ATPG?

- Software that **automatically creates test vectors** for digital circuits.
- Aims to **detect manufacturing faults** (e.g., *stuck-at*, *delay faults*).
- Ensures each fault is **activated at an input** and **observed at an output**.

ATPG

Types of ATPG

1. **Combinational ATPG**
   - Treats the circuit as purely combinational.
   - Simpler, faster, and widely used in full-scan designs.
   - Targets basic stuck-at faults.
2. **Sequential ATPG**
   - Considers memory elements (flip-flops, latches).
   - Generates sequences of vectors over time (multiple time frames).
   - Slower and more complex but handles partial-scan or non-scan designs.
3. **Multiple Time Frame ATPG**
   - Used in delay fault models.
   - Requires two or more vectors per fault (e.g., to launch and capture a transition)

ATPG

Some ATPG Fault Models

Model	Description	Targeted Faults
Stuck-at	Node is permanently at logic 0 or 1.	Static logic faults.
Transition Delay	Detects slow transitions (0⟶1 or 1⟶0).	Timing defects.
Path Delay	Verifies timing on critical paths.	Longest combinational paths.
Iddq (Current-based)	Measures abnormal current draw.	Leakage, shorts.
Bridging	Short between lines (e.g., AND-bridges).	Layout and fabrication issues.
N-detect	Detects same fault in multiple ways.	Improves defect coverage robustness

ATPG

Deterministic vs. Random ATPG

- **Random ATPG:** Fast but limited fault coverage.
- **Deterministic ATPG:** Targets specific faults with high coverage.
- **Hybrid Approaches:** Use random for easy faults, deterministic for hard faults.

Challenges

Challenge	Description
High complexity	Sequential ATPG is computationally intensive and slow.
Tool compatibility	Vector format and gate libraries must match ATPG engine expectations.
Fault coverage vs. vector count	Compact vectors vs. exhaustive fault coverage.
Integration with DFT	ATPG works best with scan, BIST, or boundary scan in place.

Outline

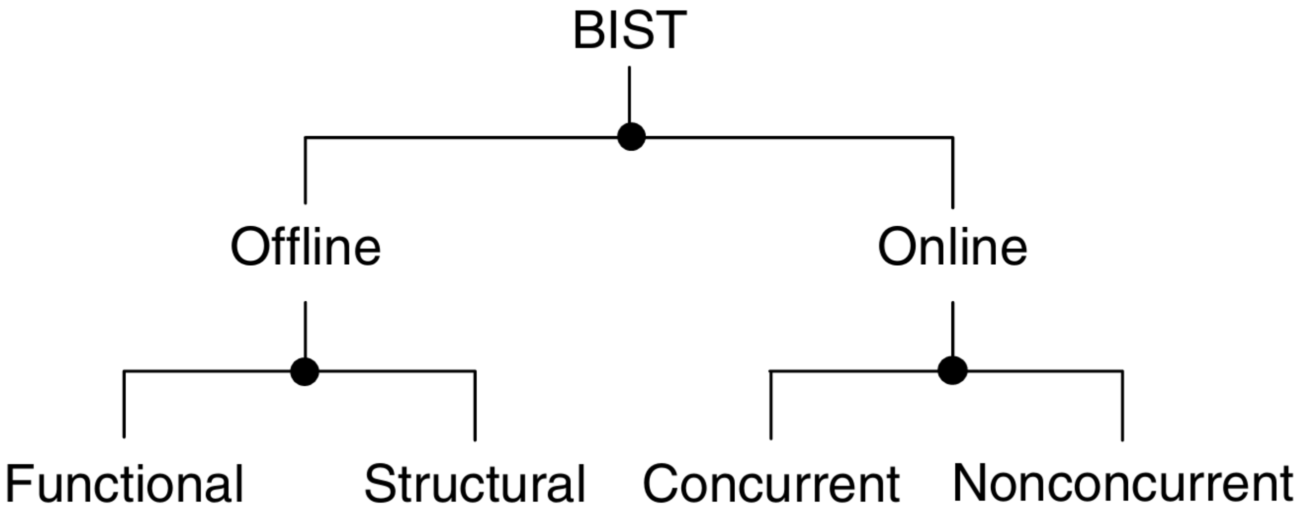
- Testing
- Introduction to DFT
- Scan
- ATPG
- BIST
- IEEE Standards for Embedded Core Testing

Logic Built-In Self-Test (BIST)

BIST is a technique in which a **portion** of a circuit on a chip, board, or system is used to test the digital logic circuit itself.

- It is crucial for **life-critical** and **mission-critical** applications.

Categories of BIST techniques



Online BIST

Online BIST is performed when the functional circuitry is in normal operational mode.

Concurrent online BIST

- The testing is conducted simultaneously during normal functional operation.
- The functional circuitry is usually implemented with coding techniques or with duplication and comparison
- When an **intermittent** or **transient** error is detected, the system will correct the error on the spot, **rollback** to its previously stored system states, and repeat the operation, or generate an interrupt signal for repeated failures.

Online BIST

Online BIST is performed when the functional circuitry is in normal operational mode.

Nonconcurrent online BIST

- The testing is performed when the functional circuitry is in idle mode.
- It is done by executing diagnosis software routines (macrocode) or diagnosis firmware routines (microcode)
- The test process can be interrupted at any time so that normal operation can resume.

Offline BIST

- Offline BIST is performed when the functional circuitry is not in normal mode.
- This technique does not detect any **real-time** errors.

Functional Offline BIST

- Performs a test on the functional specification of the circuit.
- It uses a functional or high-level failure model.
- It is implemented as diagnostic software or firmware

Offline BIST

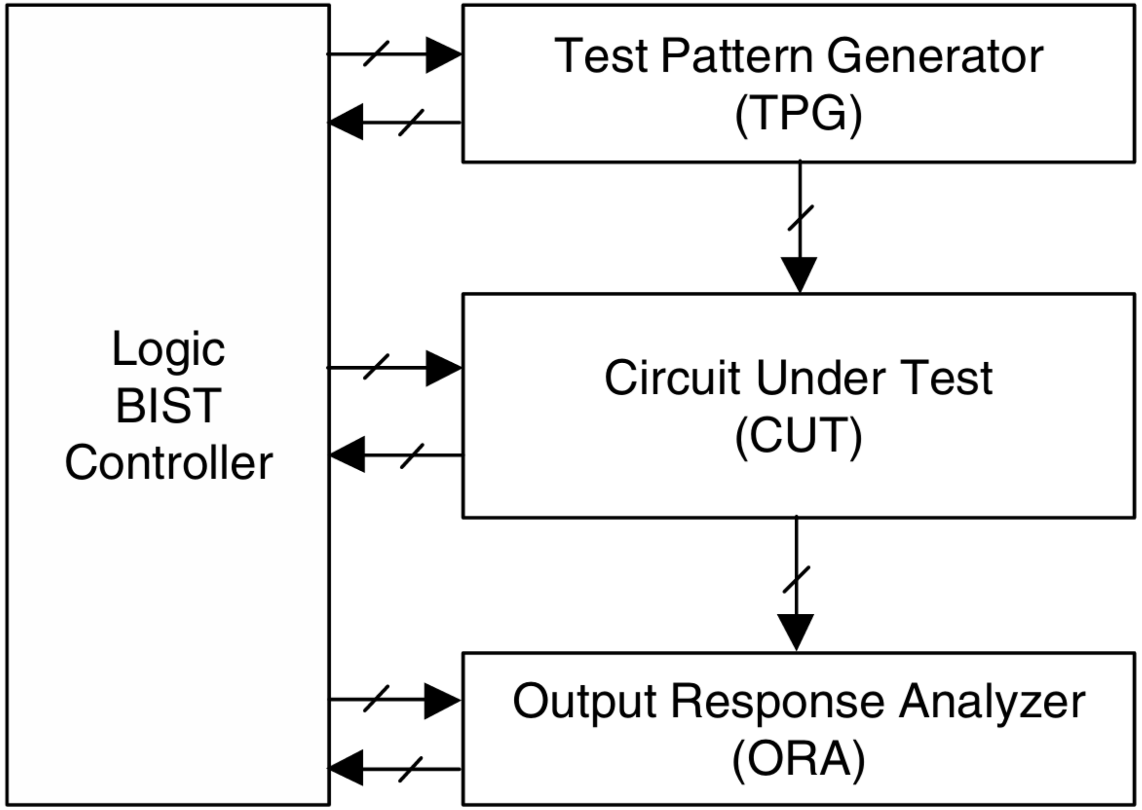
- Offline BIST is performed when the functional circuitry is not in normal mode.
- This technique does not detect any **real-time** errors.

Structural Offline BIST

- Performs a test based on the structure of the functional circuitry
    - **External BIST**
    - **Internal BIST**

BIST System

- **Test Pattern Generator (TPG):**
  - ○ Automatically generates test patterns for CUT inputs.
- **Output Response Analyzer (ORA):**
  - ○ Compacts CUT outputs into a signature.
- **BIST Controller:**
  - ○ Generates timing/control signals (scan enable, clocks).
  - ○ Coordinates TPG, CUT, and ORA operations.
  - ○ Provides pass/fail result by comparing signature vs. golden reference.
  - ○ Often includes diagnostic logic for fault analysis.



Typical logic BIST system using the structural offline BIST technique

BIST

Advantages vs. Disadvantages for Structural Offline BIST vs. Conventional Scan

Category	Advantages of BIST	Disadvantages of BIST
Test Availability	- Always available (no external tester needed).	-Requires stringent desing rules to handle X-sources (e.g., analog blocks, non-scan elements).
Error Detection	- Automatic fault detection (e.g., N-detect). - At-speed delay fault testing.	- Lower fault coverage vs. scan (may need test points).
Scan Efficiency	- Reduces test time/tester memory. - Eliminates test pattern generation costs.	- Extra area overhead for. TPG/ORA/controller.
Diagnostics	- On-chip diagnostics (trace errors to chip). - Enables in-system self-test.	- Pseudo-random patterns may miss faults.
Design Complexity	- N/A	- Challenges with: tristate buses, async signals, false/multi-cycle paths.

TEC | Tecnológico de Costa Rica

intel®