ECE 538

VLSI System Testing

Krish Chakrabarty

Design for Testability (DFT)

Outline

- Motivation and Goals
- Controllability and Observability
- Ad Hoc DFT Methods
  - Control/Test Point Insertion
  - Circuit restructuring
  - Timing considerations
- Systematic DFT Methods
- Scan Design
  - Scan cells
  - Scan chains
- Boundary Scan

Why Design for Test?

- Test generation is complex (NP-complete), limiting exact application to circuits of moderate complexity
- Random logic circuits containing, say 10^6 or more gates, or 10^3 or more flip-flops may be too large for ATPG
- Heuristic methods generally used for testing complex circuits, e.g. microprocessors or high-density RAM chips.
  - The fault coverage of such methods is low and hard to determine
- To ensure high testability, design for testability (DFT) needed.
- Make circuits easy to test by design

Testability Definitions
(Keiner 1980)

<u>Testability</u>
"A design characteristic which allows the status (operable, non-operable, or degraded) of a unit to be determined in a timely manner"

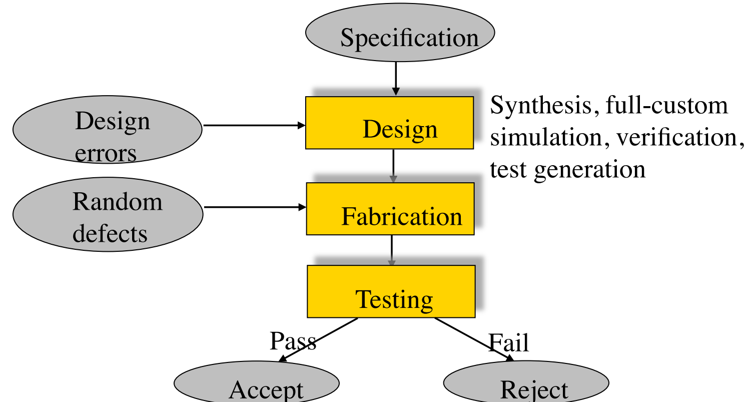
<u>Design for Testability (DFT)</u>
"A design process such that deliberate effort is expended to assure that a product may be thoroughly tested with minimum effort and cost, and that high confidence may be ascribed to the test results"

Testability Goals

- Maximize fault coverage
- Minimize test application time
- Minimize test data volume
- Minimize test generation effort
- Maximize fault resolution (isolating fault to smallest replaceable component)
- Minimize hardware/software overhead needed for testing
- Make the system as much self-testing as possible

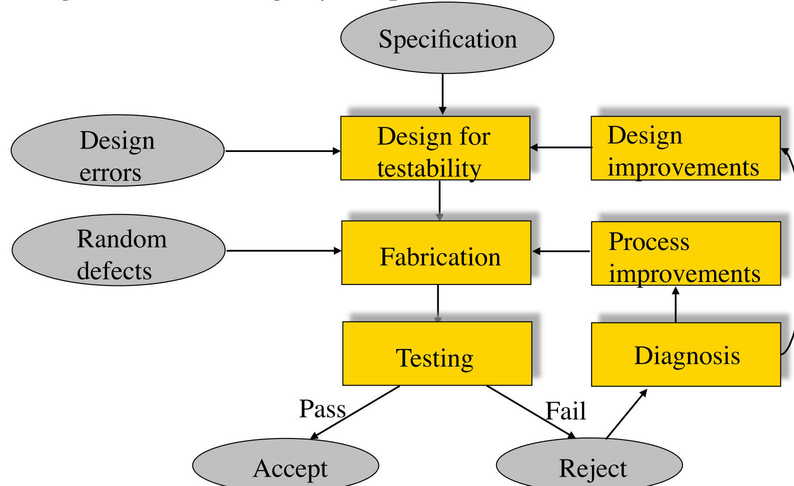
Design and Test Flow: Old View

- Test is merely an afterthought



Design and Test Flow: New View

- Design and test are tightly coupled



Definition

- *Design for testability* (DFT) refers to those design techniques that make test generation and test application cost-effective.
- DFT methods for digital circuits:
  - Ad-hoc methods
  - Structured methods:
    - *Scan*
    - *Partial Scan*
    - *Built-in self-test* (BIST)
    - *Boundary scan*
- DFT method for mixed-signal circuits:
    - Analog test bus

Ad-Hoc DFT Methods

- Good design practices learnt through experience are used as guidelines:
  - Avoid asynchronous (unclocked) feedback.
  - Make flip-flops initializable.
  - Avoid redundant gates. Avoid large fanin gates.
  - Provide test control for difficult-to-control signals.
  - Avoid gated clocks.
  - Consider ATE requirements (tristates, etc.)
- Design reviews conducted by experts or design auditing tools.
- Disadvantages of ad-hoc DFT methods:
  - Experts and tools not always available.
  - Test generation is often manual with no guarantee of high fault coverage.
  - Design iterations may be necessary.

Impact of Testing Method

Consider testing a large combinational circuit. The precise impact of each testing method depends on the circuit size, circuit complexity, and practical design constraints

		Testing method	
Testability parameter	PODEM	Exhaustive	Random
Fault coverage	good	best	worst
Test application time	good	worst	fair
Test data length	best	worst	fair
Test generation effort	worst	best	good
Suitability for self-test	bad	fair	good

Testability Factors

Controllability and Observability

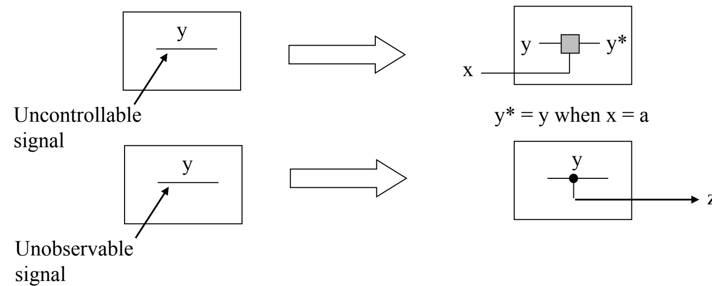
- Compute controllability and observability values for all signal nodes in the circuit
- "Bad" values can pinpoint hard-to-test areas of the circuit that need redesign
- Computation effort should be small, e.g., 10% of test generation effort. (Implies use of heuristics)

Controllability and Observability

- Several related schemes have been proposed, e.g.
  - TIMEAS (Stephenson and Grason, 1976)
  - SCOAP (Goldstein, 1979)
- Interpretation of controllability/observability values is difficult
- Often unclear how to modify circuit to improve specific controllability/observability values
- Systematic DFT can provide high levels *a priori*

Control/Test Point Insertion

- Make hard-to-control internal signals controllable via extra primary inputs and logic (control points)
- Make hard-to-observe internal signals observable via extra primary outputs and logic (test points)



Control/Test Points

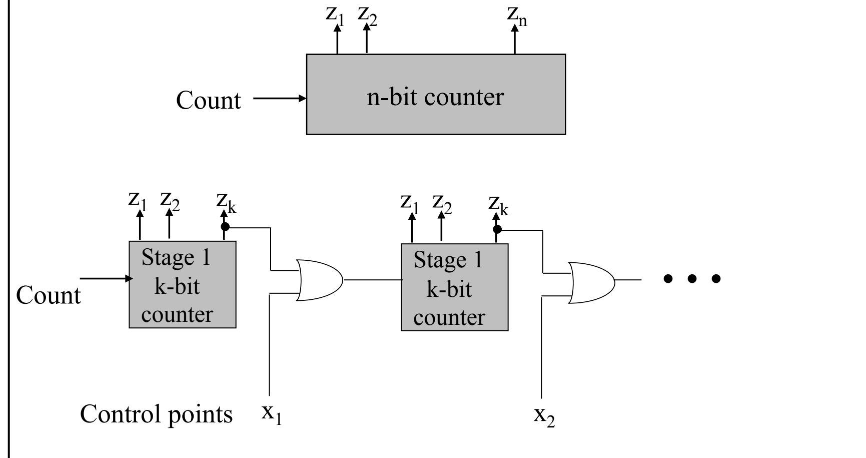
Site Selection

- Memory elements determining a system's control states
- Long narrow subcircuits with buried components
- Short wide subcircuits with high fan-in or fan-out
- Internal system buses
- Redundant circuits containing undetectable faults
- Circuit bottlenecks determined by programs such as SCOAP

Major Limitations

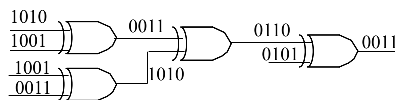
- Availability of spare I/O pins
- Cost of added test/control circuits

Circuit Restructuring



XOR-Based Designs

- Circuits composed exclusively of XOR (and XNOR) are easy to test
  - Easy to observe
  - Easy to control
- Every all-XOR/XNOR circuit composed of 2-input XOR gates can be tested for all SSL faults with 3 test patterns
- Every all-XOR/XNOR circuit composed of 2-input XOR gates can be tested for all cell faults with 4 test patterns

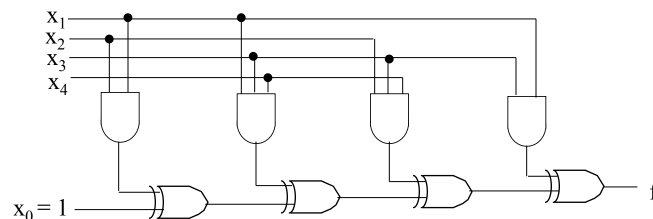


Reed-Muller Circuits

- Every Boolean function can be expressed in the "Reed-Muller" form consisting of an XOR sum of product terms with no complementation

Example:

$f(x_1,x_2,x_3,x_4) = 1 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_2x_3 \oplus x_2x_3x_4$



Reed-Muller Circuits

- An n-input Reed-Muller circuit can be tested for all SSL faults using n+4 tests patterns
- It requires an extra AND gate, a control input ans a test point
- The number of gates and the circuit delay grows exponentially with n.

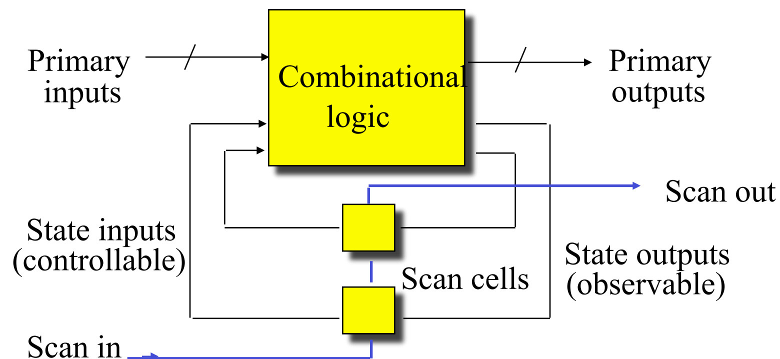
$f = a + b$
$= ab + \bar{a}b + a\bar{b}$
$= ab \oplus \bar{a}b \oplus a\bar{b}$ (sum of minterms)
$= ab \oplus (1 \oplus a)b \oplus a(1 \oplus b)$
$= ab \oplus b \oplus ab \oplus a \oplus ab = a \oplus b \oplus ab$

Design Rules Summary

- Design controllable (e.g. initializable) and controllable circuits with careful selection of control/test points
- Partition large hard-to-test circuits into small testable components
- Allow feedback paths to be opened and closed
- Avoid redundancy, or allow it to be overridden during testing
- Avoid asynchronous circuits and provide access to clock signals
- ***Often ad hoc design modification is too late to significantly improve a circuit's testability***

Scan Design

- Make all flip-flops directly controllable and observable by adding multiplexers
- Popular design-for-test (DFT) technique-circuit is now combinational for testing purposes



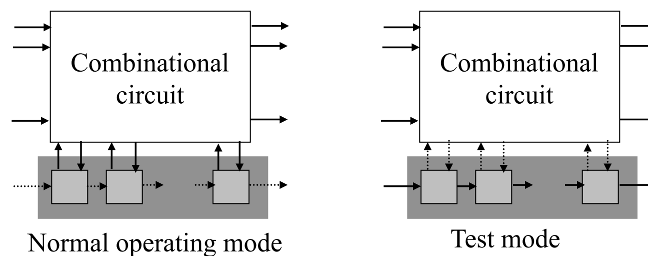
Scan Design

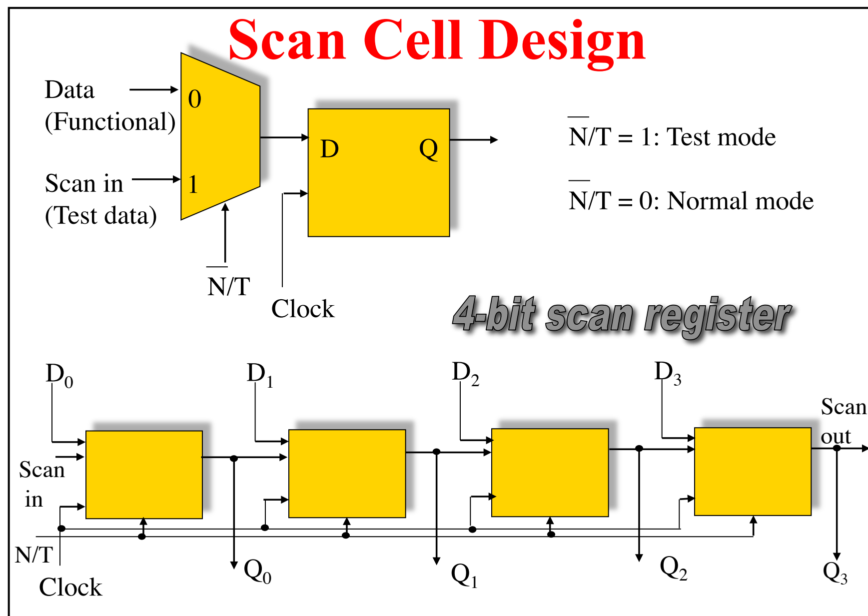
- Circuit is designed using pre-specified design rules.
- Test structure (hardware) is added to the verified design:
  - Add a *test control* (TC) primary input.
  - Replace flip-flops by *scan flip-flops* (SFF) and connect to form one or more shift registers in the test mode.
  - Make input/output of each scan shift register controllable/ observable from PI/PO.
- Use combinational ATPG to obtain tests for all testable faults in the combinational logic.
- Add shift register tests and convert ATPG tests into scan sequences for use in manufacturing test.

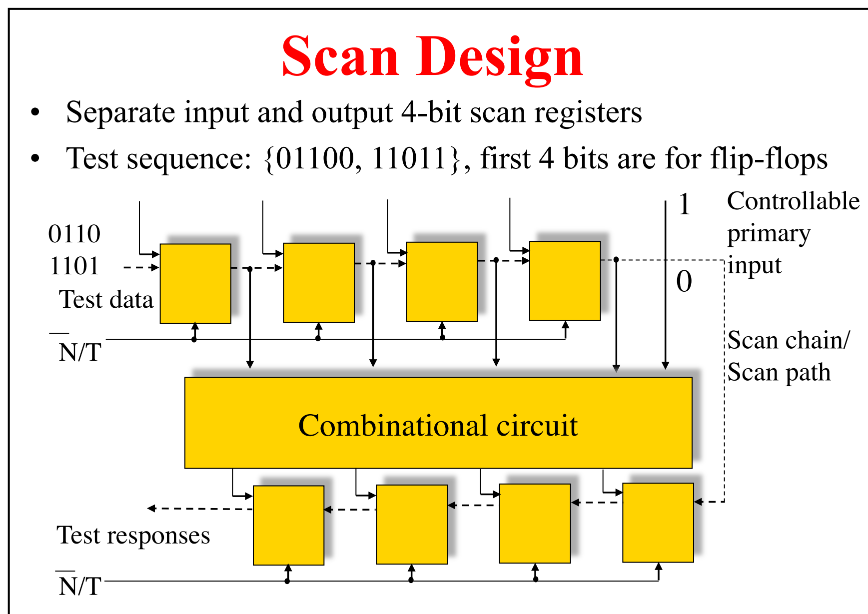
Scan Design

Concept

- Memory elements (latches and flip-flops) are designed so that they can be reconfigured dynamically to form a shift register R during testing
- Test data transferred serially to and from R making memory state completely controllable and observable



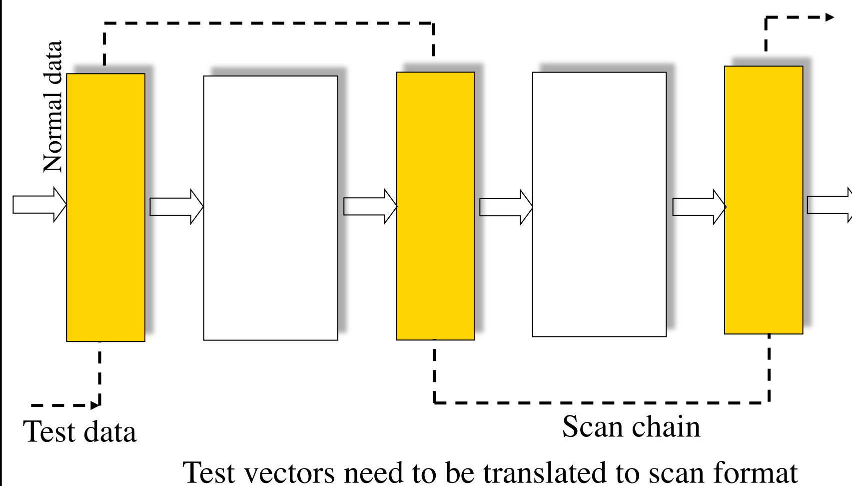

Scan Cell Design
Data
(Functional)
0
1
Scan in
(Test data)
N/T
D
Q
Clock
N/T = 1: Test mode
N/T = 0: Normal mode
4-bit scan register
D0
D1
D2
D3
Scan
in
N/T
Clock
Scan
out
Q0
Q1
Q2
Q3
ECE 538
Krish Chakrabarty
23


Scan Design
• Separate input and output 4-bit scan registers
• Test sequence: {01100, 11011}, first 4 bits are for flip-flops
0110
1101
Test data
N/T
1
0
Controllable
primary
input
Scan chain/
Scan path
Combinational circuit
Test responses
N/T
ECE 538
Krish Chakrabarty
24

Steps in Scan Testing

- $\overline{N}/T = 1$: Scan in test pattern, hold appropriate bit pattern on controllable primary inputs
- $\overline{N}/T = 0$: Apply test pattern to combinational circuit
- $\overline{N}/T=1$: Scan out test responses
- Scan provides complete controllability and observability
- Testing time? How many cycles? How to test scan registers?

Long Scan Chains



Test vectors need to be translated to scan format

Comments on Scan Design

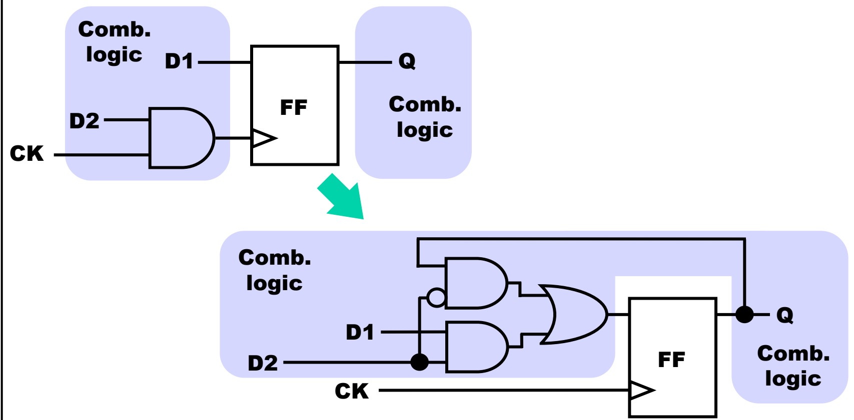
- Allows complete controllability and observability
- Test pattern must be generated primarily for the combinational circuit
- Hardware overhead is small: a few extra pins and some (5 to 20%) extra logic for the latches and flip-flops
- Test application is slow
- Limited to a few hundred memory latches

Scan Design Rules

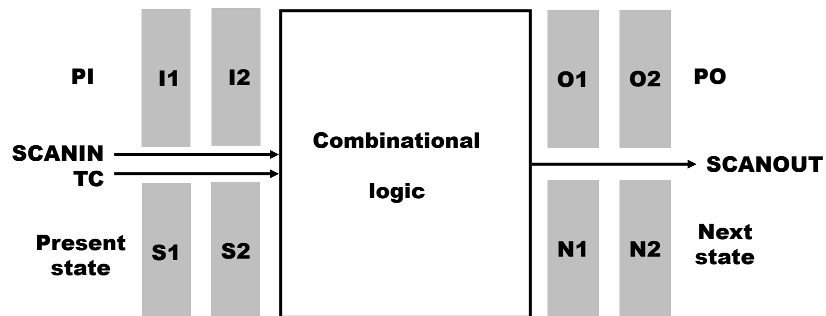
- Use only clocked D-type of flip-flops for all state variables.
- At least one PI pin must be available for test; more pins, if available, can be used.
- All clocks must be controlled from PIs.
- Clocks must not feed data inputs of flip-flops.

Correcting a Rule Violation

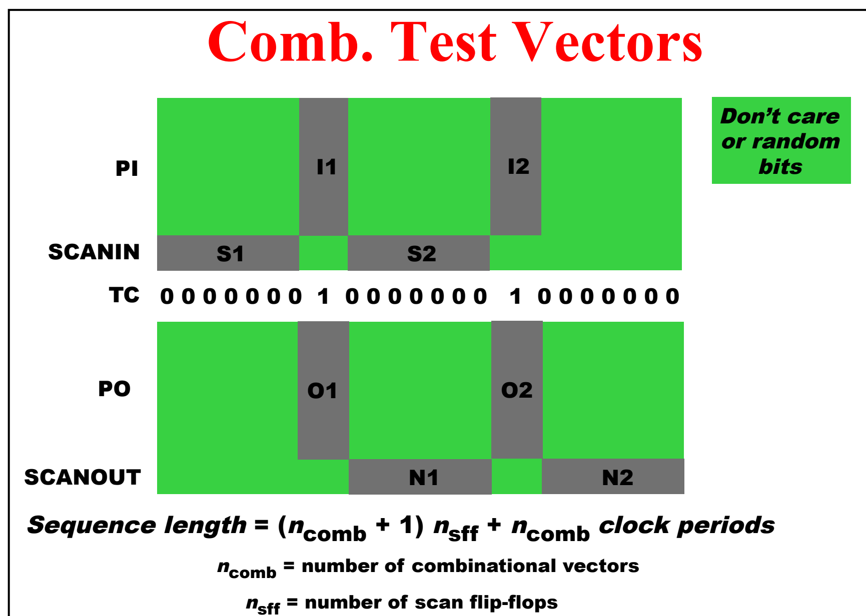
- All clocks must be controlled from PIs.



Comb. Test Vectors



Comb. Test Vectors



Sequence length = (n_{comb} + 1) n_{sff} + n_{comb} *clock periods*

n_{comb} = number of combinational vectors

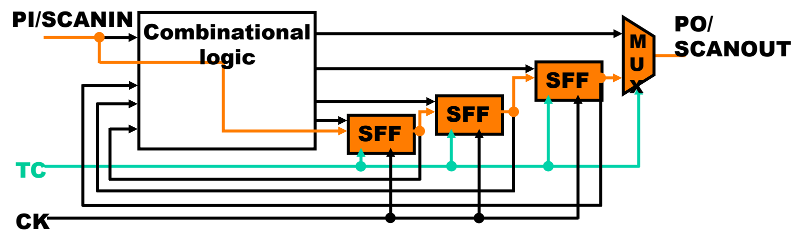
n_{sff} = number of scan flip-flops

Testing Scan Register

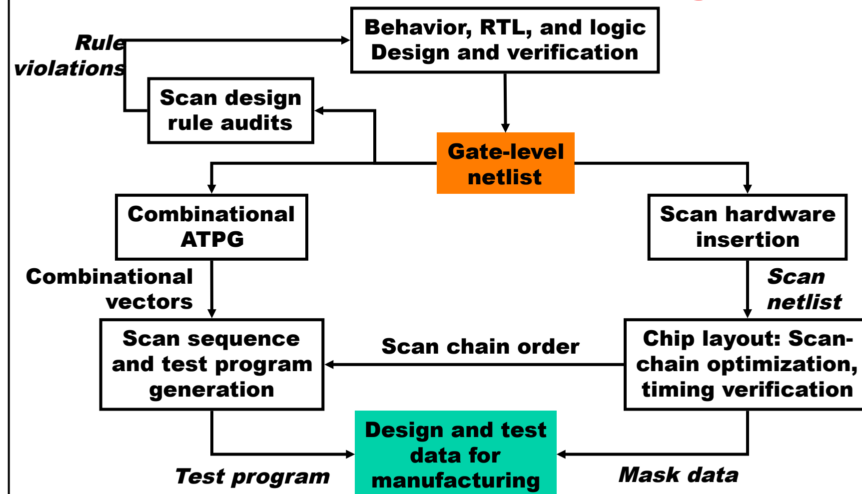
- Scan register must be tested prior to application of scan test sequences.
- A shift sequence 00110011 . . . of length n_{sff}+4 in scan mode (TC=0) produces 00, 01, 11 and 10 transitions in all flip-flops and observes the result at SCANOUT output.
- Total scan test length: $(n_{comb} + 2)\, n_{sff} + n_{comb} + 4$ *clock periods*.
- Example: 2,000 scan flip-flops, 500 comb. vectors, total scan test length ~ 10^6 clocks.
- Multiple scan registers reduce test length.

Multiple Scan Registers

- Scan flip-flops can be distributed among any number of shift registers, each having a separate *scanin* and *scanout* pin.
- Test sequence length is determined by the longest scan shift register.
- Just one *test control* (TC) pin is essential.



Automated Scan Design



Timing and Power

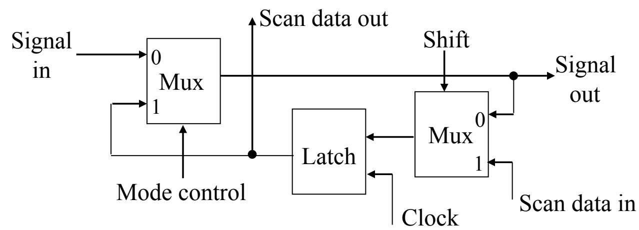
- Small delays in scan path and clock skew can cause race condition.
- Large delays in scan path require slower scan clock.
- Dynamic multiplexers: Skew between TC and $\overline{TC}$ signals can cause momentary shorting of D and SD inputs.
- Random signal activity in combinational circuit during scan can cause excessive power dissipation.

Summary

- Scan is the most popular DFT technique:
  - Rule-based design
  - Automated DFT hardware insertion
  - Combinational ATPG
- Advantages:
  - Design automation
  - High fault coverage; helpful in diagnosis
  - Hierarchical – scan-testable modules are easily combined into large scan-testable systems
  - Moderate area (~10%) and speed (~5%) overheads
- Disadvantages:
  - Large test data volume and long test time
  - Basically a slow speed (DC) test

Boundary Scan

- IEEE standard 1149.1 for incorporating scan design into chips and boards
- Shift latch placed at each pin
- Originally envisaged for PCBs, but also applicable to MCMs and core-based systems-on-a-chip



Boundary Scan