第3讲 可测试性设计(2)

李晓维

中科院计算技术研究所

Email: lxw@ict.ac.cn

http://people.ucas.edu.cn/~lxw

回顾第1讲

Test Generation

- A test is a sequence of test patterns, called test vectors, applied to the CUT whose outputs are monitored and analyzed for the correct response
  - Exhaustive testing – applying all possible test patterns to CUT
  - Functional testing – testing every truth table entry for a combinational logic CUT
    - Neither of these are practical for large CUTs
- Fault coverage is a quantitative measure of quality of a set of test vectors

回顾第1讲

Test Generation

- *Goal*: find efficient set of test vectors with maximum fault coverage
- Fault simulation used to determine fault coverage
  - Requires fault models to emulate behavior of defects
- A good fault model:
  - Is computationally efficient for simulation
  - Accurately reflects behavior of defects
- No single fault model works for all possible defects

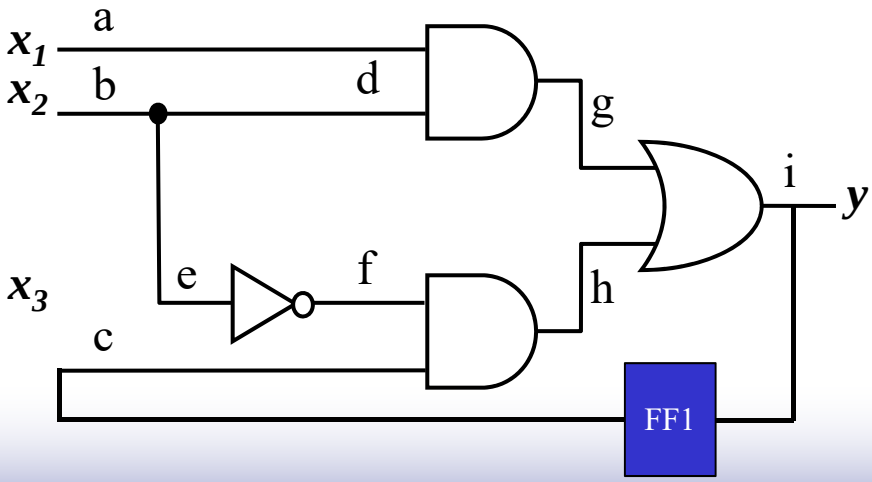
回顾第1讲

Fault Models

- A given fault model has k types of faults
  - k = 2 for most fault models
- A given circuit has n possible fault sites
- Multiple fault model –circuit can have multiple faults (including single faults)
  - Number of multiple fault = $(k+1)^n-1$
    - Each fault site can have 1-of-k fault types or be fault-free
    - The "-1" represents the fault-free circuit
  - Impractical for anything but very small circuits
- Single fault model – circuit has only 1 fault
  - Number of single faults = $k \times n$
  - Good single fault coverage generally implies good multiple fault coverage

Stuck-at Faults

- Any line can be
  - Stuck-at-0/1 (SA0/SA1)
- Example circuit:
  - # fault sites: n=9
  - # single faults =2×9=18
  - # collapsed faults = $2\times(P_O+F_O)+G_I-N_I=10$



Truth table for fault-free behavior and behavior of all possible stuck-at faults

$x_1x_2x_3$	000	001	010	011	100	101	110	111
y	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1
b SA0	0	1	0	1	0	1	0	1
b SA1	0	0	0	0	1	1	1	1
c SA0	0	0	0	0	0	0	1	1
c SA1	1	1	0	0	1	1	1	1
d SA0	0	1	0	0	0	1	0	0
d SA1	0	1	0	0	1	1	1	1
e SA0	0	1	0	1	0	1	1	1
e SA1	0	0	0	0	0	0	1	1
f SA0	0	0	0	0	0	0	1	1
f SA1	0	1	0	1	0	1	1	1
g SA0	0	1	0	0	0	1	0	0
g SA1	1	1	1	1	1	1	1	1
h SA0	0	0	0	0	0	0	1	1
h SA1	1	1	1	1	1	1	1	1
i SA0	0	0	0	0	0	0	0	0
i SA1	1	1	1	1	1	1	1	1

Chapter 2

Design for Testability

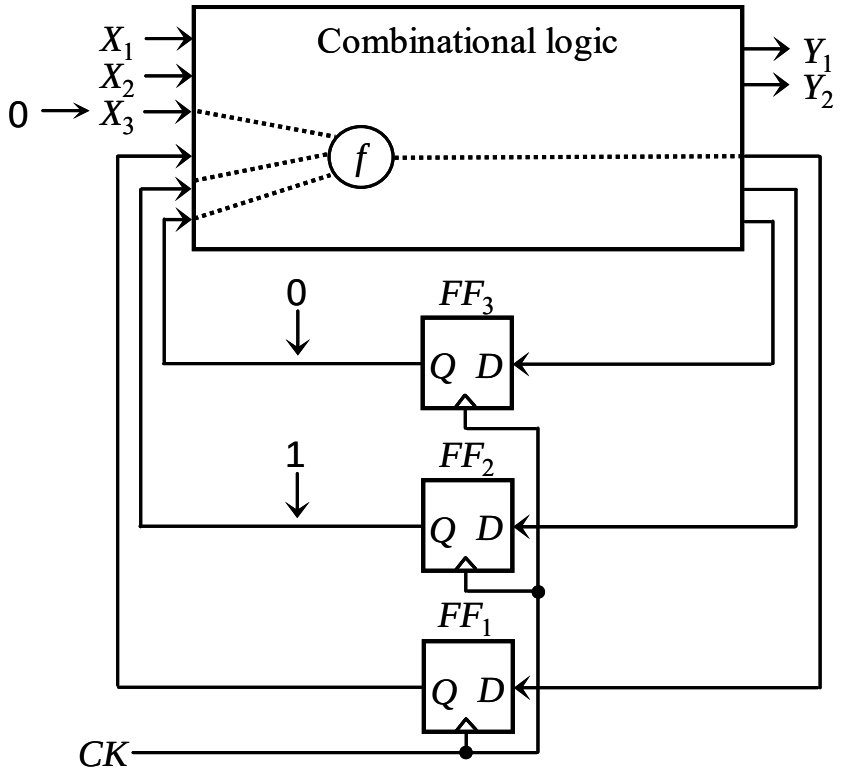
Design For Testability - contents

- Introduction
- Testability Analysis
- Design for Testability Basics
- Scan Cell Designs
- Scan Architectures
- Scan Design Rules
- Scan Design Flow
- Special-Purpose Scan Designs
- RTL Design for Testability
- Concluding Remarks

Structured Approach

- Scan design
  - Convert the sequential design into a scan design
  - Three modes of operation
    - Normal mode
      - All test signals are turned off
      - The scan design operates in the original functional configuration
    - Shift mode
    - Capture mode
      - In both shift and capture modes, a test mode signal *TM* is often used to turn on all test-related fixes

Structured Approach - Scan Design

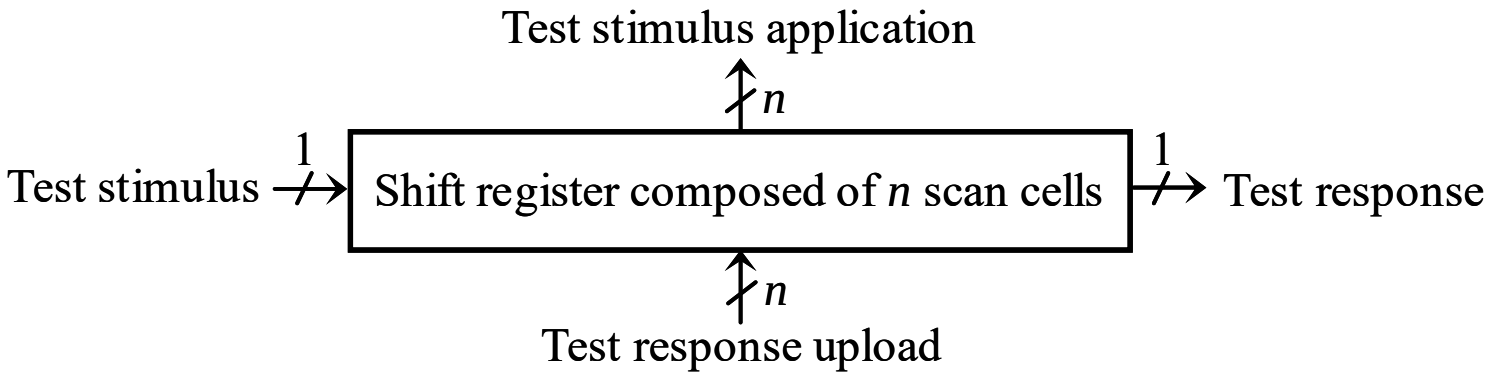


Assume that a stuck-at fault *f* in the combinational logic requires the primary input *X*3, flip-flop FF_2, and flip-flop FF_3, to be set to 0, 1, and 0.

The main difficulty in testing a sequential circuit stems from the fact that it is difficult to control and observe the internal state of the circuit.

Difficulty in testing a sequential circuit

Structured Approach - Scan Design



- Converting selected storage elements in the design into **scan cells**.
- Stitching them together to form **scan chains**.

How to detect stuck-at fault f :
(1) switching to shift mode and shifting in the desired test stimulus, 1 and 0, to FF_2 and FF_3, respectively
(2) driving a 0 onto primary input X_3
(3) switching to capture mode and applying one clock pulse to capture the fault effect into FF_1
(4) switching back to shift mode and shifting out the test response stored in FF_1, FF_2, and FF_3 for comparison with the expected response.

Scan Cell Design

- A scan cell has two inputs: data input and scan input
  - In normal/capture mode, data input is selected to update the output
  - In shift mode, scan input is selected to update the output
- Three widely used scan cell designs
  - Muxed-D Scan Cell
  - Clocked-Scan Cell
  - LSSD Scan Cell

Comparing three scan cell designs

	Advantages	Disadvantages
Muxed-D Scan Cell	Compatibility to modern designs Comprehensive support provided by existing design automation tools	Add a multiplexer delay
Clocked-Scan Cell	No performance degradation	Require additional shift clock routing
LSSD Scan Cell	Insert scan into a latch-based design Guarantee to be race-free	Increase routing complexity

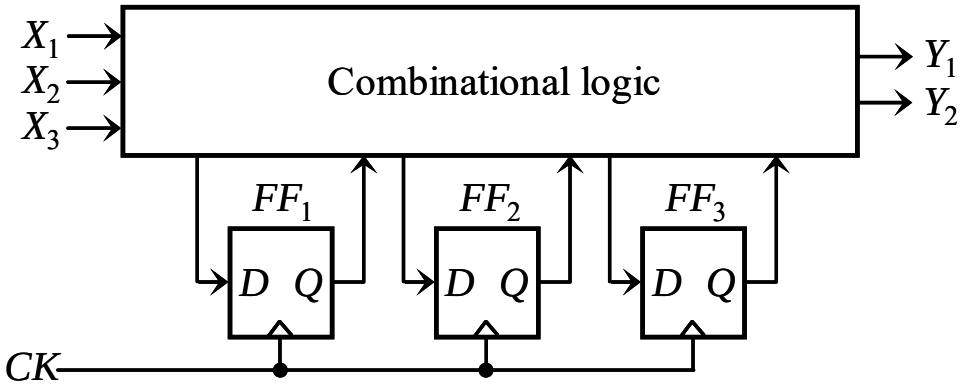
Scan Architectures

- Full-Scan Design
  - All or almost all storage element are converted into scan cells and combinational ATPG is used for test generation
- Partial-Scan Design
  - A subset of storage elements are converted into scan cells and sequential ATPG is typically used for test generation
- Random-Access Scan Design
  - A random addressing mechanism, instead of serial scan chains, is used to provide direct access to read or write any scan cell

Full-Scan Design

- All storage elements are replaced with scan cells
  - All inputs can be controlled
  - All outputs can be observed
- Advantage:
  - Converts sequential ATPG into combinational ATPG
- Almost full-scan design
  - A small percentage of storage elements are not replaced with scan cells
    - For performance reasons
      - Storage elements that lie on critical paths
    - For functional reasons
      - Storage elements driven by a small clock domain that are deemed too insignificant to be worth the additional scan insertion effort

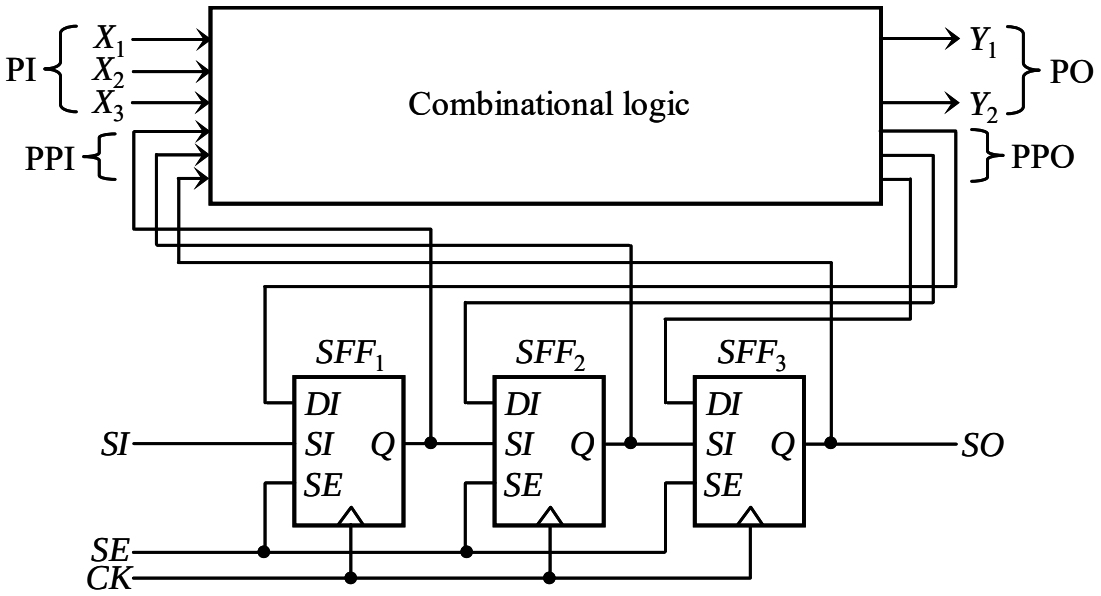
Muxed-D Full-Scan Design



Sequential circuit example

The three D flip-flops, FF_1, FF_2 and FF_3, are replaced with three muxed-D scan cells, SFF_1, SFF_2 and SFF_3, respectively.

Muxed-D Full-Scan Design



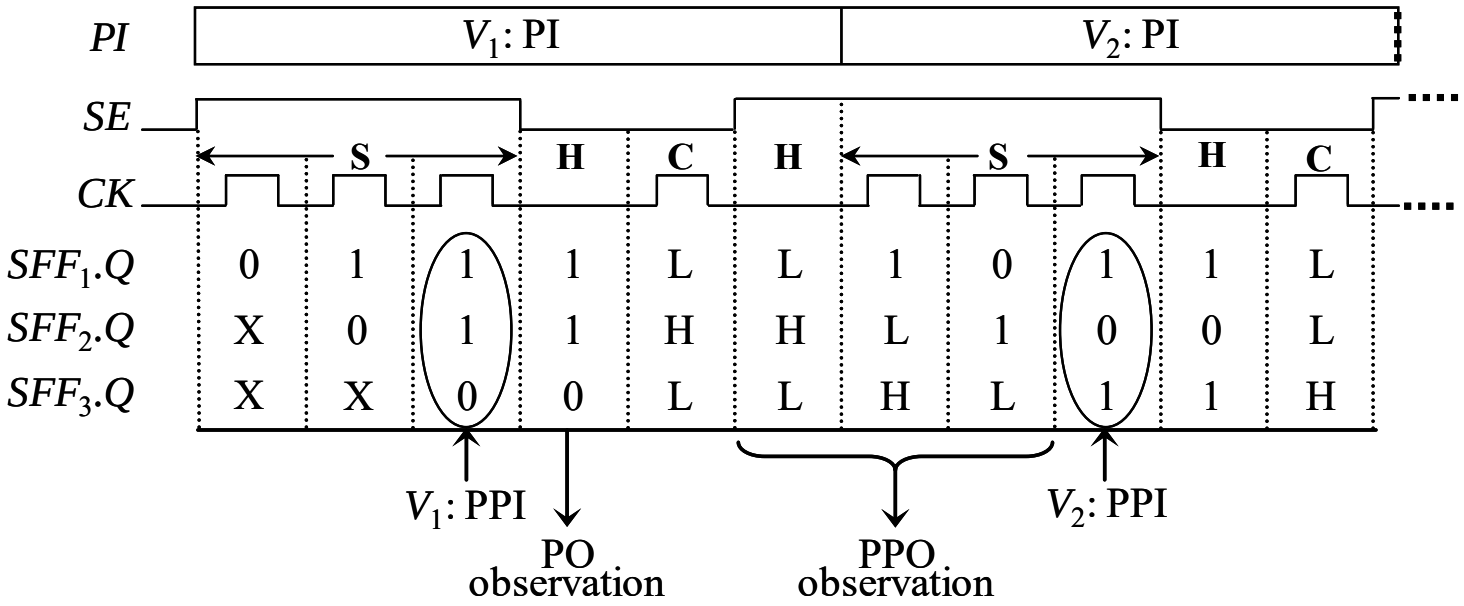
(a) Muxed-D full-scan circuit

To form a scan chain, the scan input *SI* of SFF_2 and SFF_3 are connected to the output *Q* of the previous scan cell, SFF_1 and SFF_2, respectively. In addition, the scan input *SI* of the first scan cell SFF_1 is connected to the primary input *SI*, and the output *Q* of the last scan cell SFF_3 is connected to the primary output *SO*.

Muxed-D Full-Scan Design

- *Primary inputs* (PIs)
  - the external inputs to the circuit
  - can be set to any required logic values
  - set directly in parallel from the external inputs
- *Pseudo primary inputs* (PPIs)
  - the scan cell outputs
  - can be set to any required logic values
  - are set serially through scan chain inputs
- *Primary outputs* (POs)
  - the external outputs of the circuit
  - can be observed
  - are observed directly in parallel from the external outputs
- *Pseudo primary outputs* (PPOs)
  - the scan cell inputs
  - can be observed
  - are observed serially through scan chain outputs

Muxed-D Full-Scan Design



S: shift operation / **C**: capture operation / **H**: hold cycle

Number of scan-test cycles = (Nvector + 1) Nsff + Nvector

(b) Test operations

Muxed-D Full-Scan Design

Circuit Operation type	Scan cell mode	*TM*	SE
Normal	Normal	0	0
Shift Operation	Shift	1	1
Capture Operation	Capture	1	0

Circuit operation type and scan cell mode

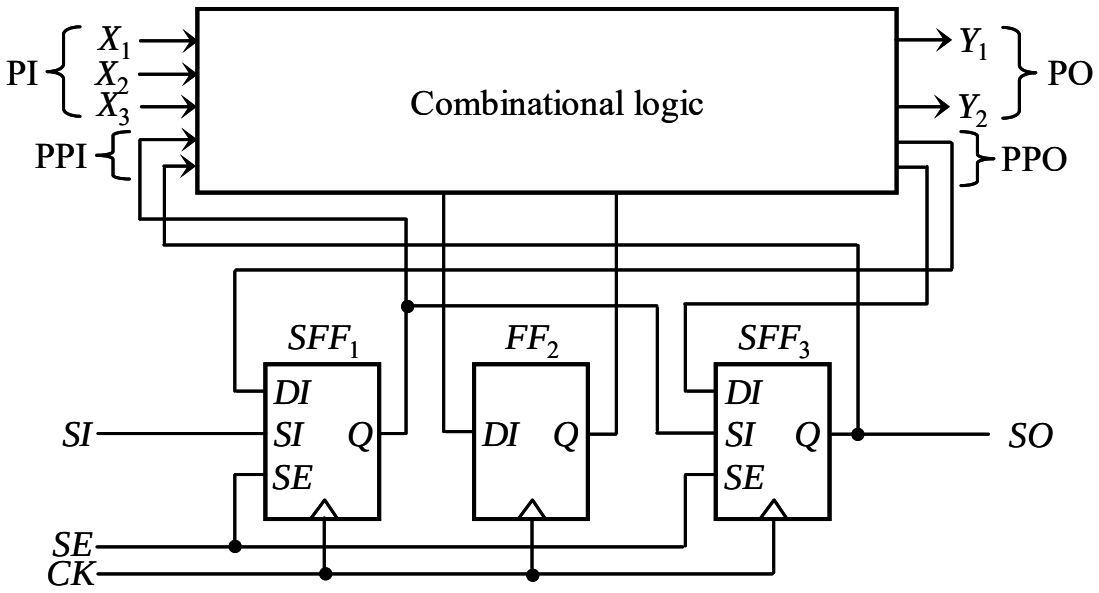
ATPG Example: S5378

	Original	Full-scan
Number of combinational gates	2,781	2,781
Number of non-scan flip-flops (10 gates each)	179	0
Number of scan flip-flops (14 gates each)	0	179
Gate overhead	0.0%	15.66%
Number of faults	4,603	4,603
PI/PO for ATPG	35/49	214/228
Fault coverage	70.0%	99.1%
Fault efficiency	70.9%	100.0%
CPU time on SUN Ultra II, 200MHz processor	5,533 s	5 s
Number of ATPG vectors	414	585
Scan sequence length	414	105,662

Partial-Scan Design

- Was once used in the industry long before full-scan design became the dominant scan architecture.
- Can also be implemented using muxed-D scan cells, clocked-scan cells, or LSSD scan cells.
- Either combinational ATPG or sequential ATPG can be used.

Partial-Scan Design



An example of muxed-D partial-scan design

A scan chain is constructed with two scan cells SFF_1 and SFF_3, while flip-flop FF_2 is left out.

It is possible to reduce the test generation complexity by splitting the single clock into two separate clocks, one for controlling all scan cells, the other for controlling all non-scan storage elements. However, this may result in additional complexity of routing two separate clock trees during physical implementation.

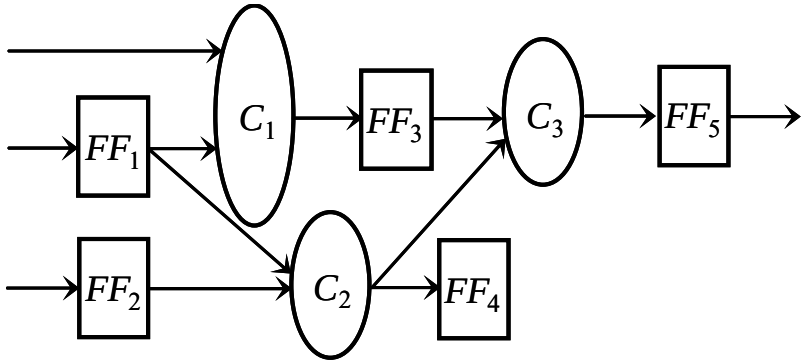
Partial-Scan Design

- Scan cell selection
  - A functional partitioning approach
    - A circuit is composed of a data path portion and a control portion
    - Storage elements on the data path are left out of the scan cell replacement process
    - Storage elements on the control path can be replaced with scan cells
  - A pipelined or feed-forward partial-scan design approach
    - Make the sequential circuit feedback-free by selecting the storage elements to break all sequential feedback loops
    - First construct a structure graph for the sequential circuit
  - A balanced partial-scan design approach
    - Use a target sequential depth to simplify the test generation process for the pipelined or feed-forward partial-scan design

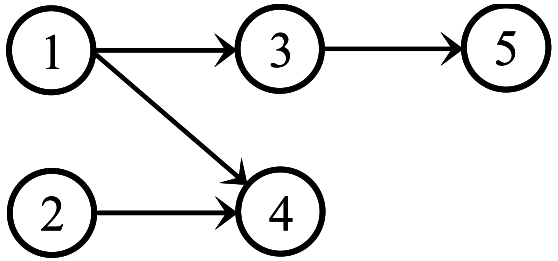
Partial-Scan Design - Structure Graph

- A feedback-free sequential circuit
  - Use a *directed acyclic graph* (DAG)
  - The maximum level in the structure graph is referred to as **sequential depth**
- A sequential circuit containing feedback loops
  - Use a *directed cyclic graph* (DCG)

Sequential circuit and its structure graph



(a) Sequential Circuit



(b) Structure graph
Sequential depth is 3

The sequential depth of a circuit is equal to the maximum number of clock cycles that needs to be applied in order to control and observe values to and from all non-scan storage elements

- The sequential depth of a full-scan circuit is 0

Partial-Scan Design

- Advantage:
  - Reduce silicon area overhead
  - Reduce performance degradation
- Disadvantage:
  - Can result in lower fault coverage
  - Longer test generation time
  - Offers less support for debug, diagnosis and failure analysis

Partial Scan Example

- Circuit: TLC
- 355 gates
- 21 flip-flops

Scan flip-flops	Max. cycle length	Depth*	ATPG CPU s	Fault sim. CPU s	Fault cov.	ATPG vectors	Test seq. length
0	4	14	1,247	61	89.01%	805	805
4	2	10	157	11	95.90%	247	1,249
9	1	5	32	4	99.20%	136	1,382
10	1	3	13	4	100.00%	112	1,256
21	0	0	2	2	100.00%	52	1,190

* Cyclic paths ignored

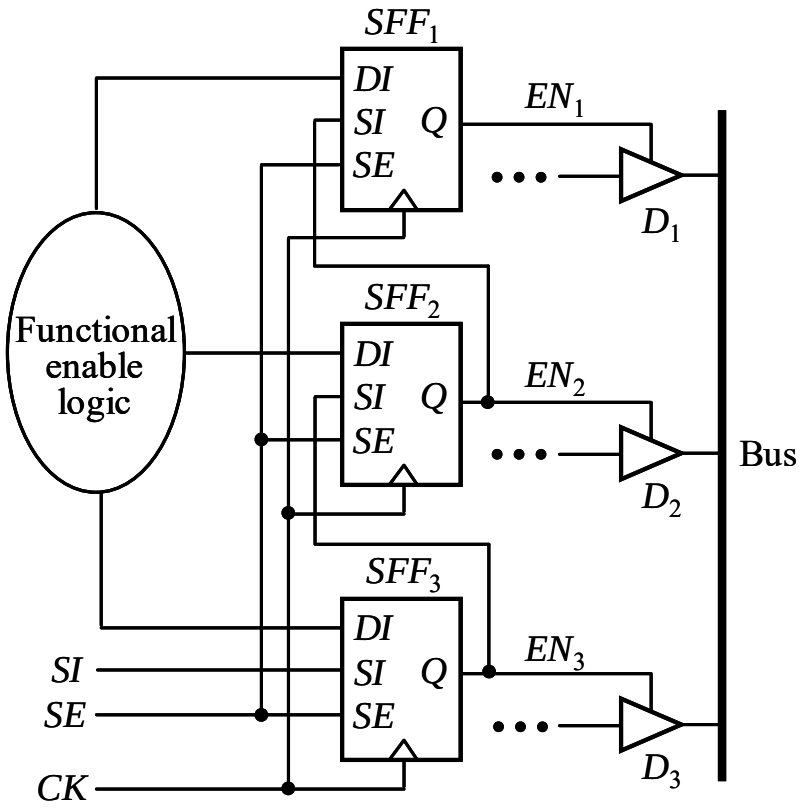
Partial vs. Full Scan: S5378

	Original	Partial-scan	Full-scan
Number of combinational gates	2,781	2,781	2,781
Number of non-scan flip-flops (10 gates each)	179	149	0
Number of scan flip-flops (14 gates each)	0	30	179
Gate overhead	0.0%	2.63%	15.66%
Number of faults	4,603	4,603	4,603
PI/PO for ATPG	35/49	65/79	214/228
Fault coverage	70.0%	93.7%	99.1%
Fault efficiency	70.9%	99.5%	100.0%
CPU time on SUN Ultra II 200MHz processor	5,533 s	727 s	5 s
Number of ATPG vectors	414	1,117	585
Scan sequence length	414	34,691	105,662

Scan Design Rules

Design Style	*Scan Design Rule*	*Recommended Solution*
Tri-state buses	Avoid during shift	Fix bus contention during shift
Bi-directional 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 pin(s)
Clocks driving data	Avoid	Block clocks to the data portion
Floating buses	Avoid	Add bus keepers
Floating inputs	Not recommended	Tie to Vcc 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

Tri-State Buses



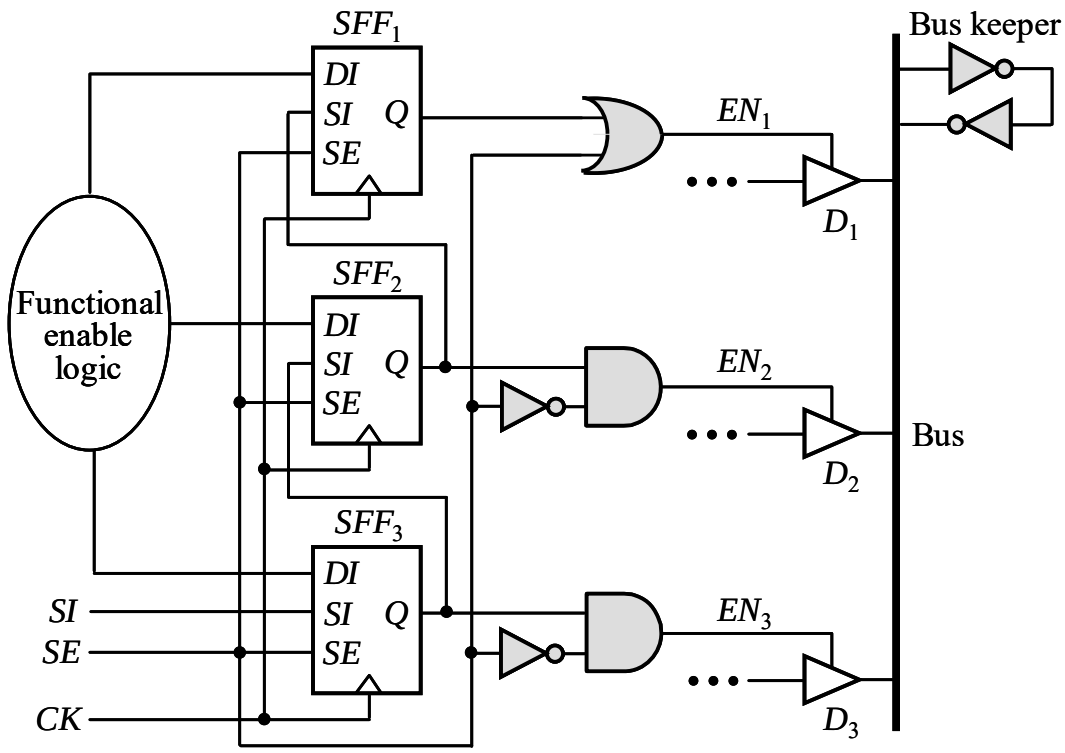
Original Circuit

Bus contention occurs when two bus drivers force opposite logic values onto a tri-state bus.

Bus contention is designed not to happen during the normal operation, and is typically avoided during the capture operation.

However, during the shift operation, no such guarantees can be made.

Tri-State Buses

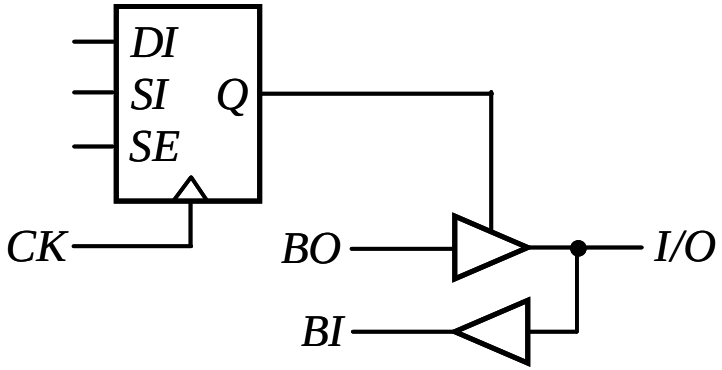


Modified circuit fixing bus contention

EN_1 is forced to 1 to enable the D_1 bus driver, while EN_2 and EN_3 are set to 0 to disable both D_2 and D_3 bus drivers, when $SE = 1$.

A bus without a pull-up, pull-down, or bus keeper may result in fault coverage loss, the bus keeper is added.

Bi-Directional I/O Ports

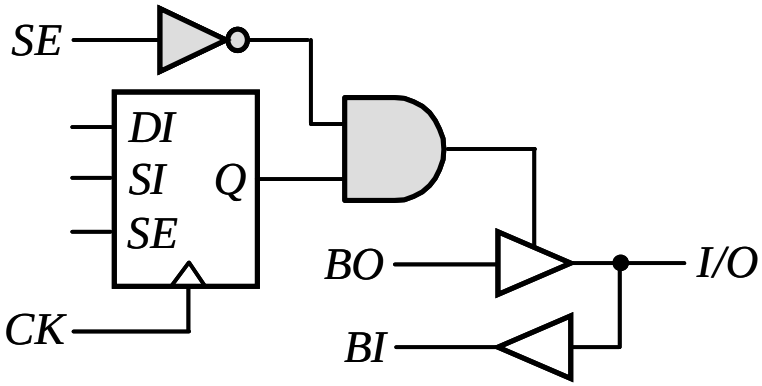


(a) Original circuit

Conflicts may occur at a bidirectional I/O port during the shift operation.

Since the output value of the scan cell can vary during the shift operation, the output tri-state buffer may become active, resulting in a conflict if BO and the I/O port driven by the tester have opposite logic values.

Bi-Directional I/O Ports

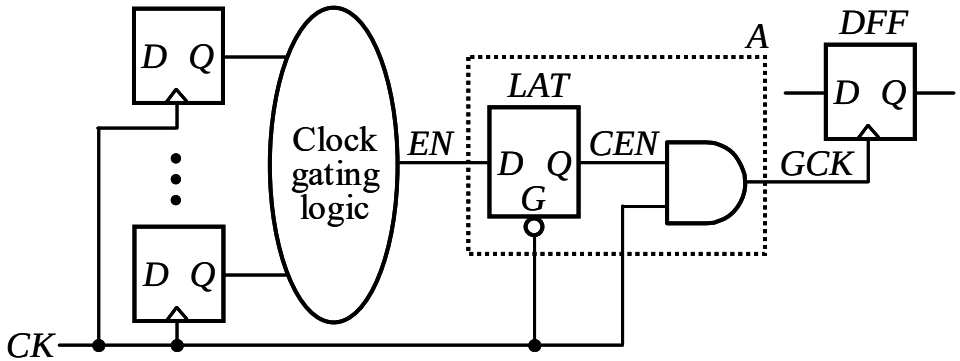


(b) Modified circuit

Fix this problem by forcing the tri-state buffer to be inactive when *SE* = 1, and the tester is used to drive the I/O port during the shift operation.

During the capture operation, the applied test vector determines whether a bi-directional I/O port is used as input or output and controls the tester appropriately.

Gated Clocks

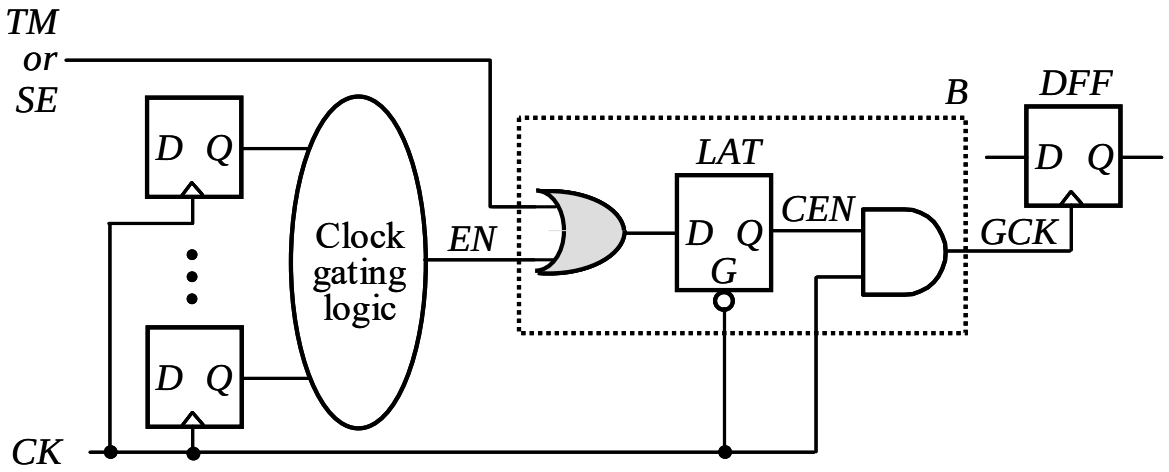


(a) Original circuit

Although clock gating is a good approach for reducing power consumption, it prevents the clock ports of some flip-flops from being directly controlled by primary inputs.

Gated Clocks

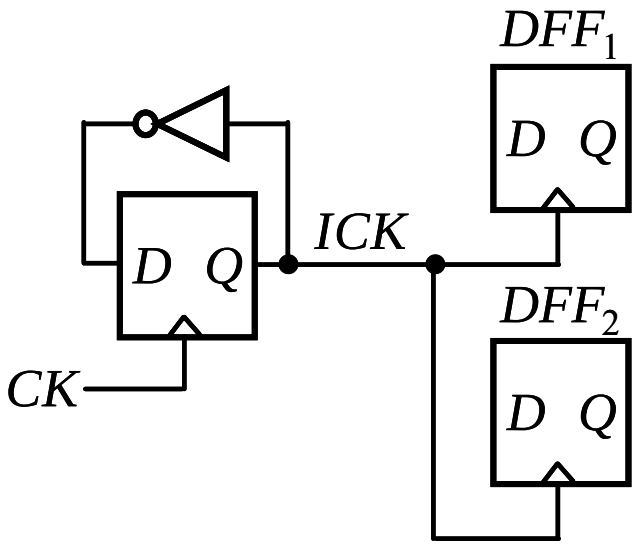
The clock gating function should be disabled at least during the shift operation.



(b) Modified Circuit

An OR gate is used to force *CEN* to 1 using either the test mode signal *TM* or the scan enable signal *SE*.

Derived clocks

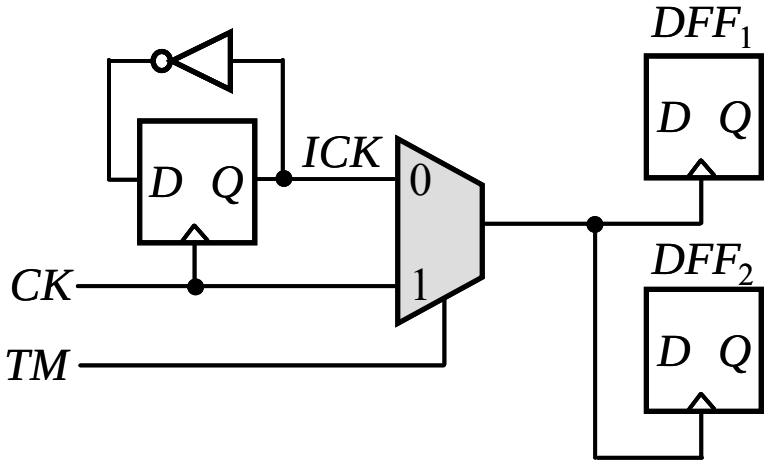


(a) Original circuit

A derived clock is a clock signal generated internally from a storage element or a clock generator.

These clock signals need to be bypassed during the entire test operation.

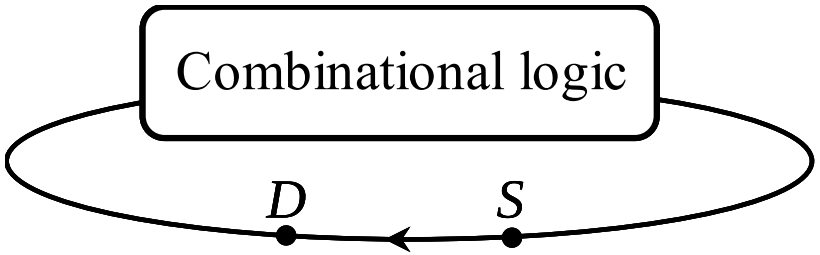
Derived clocks



(b) Modified circuit

A multiplexer selects *CK*, which is a clock directly controllable from a primary input, to drive DFF_1 and DFF_2, during the entire test operation, when $TM = 1$.

Combinational Feedback Loops



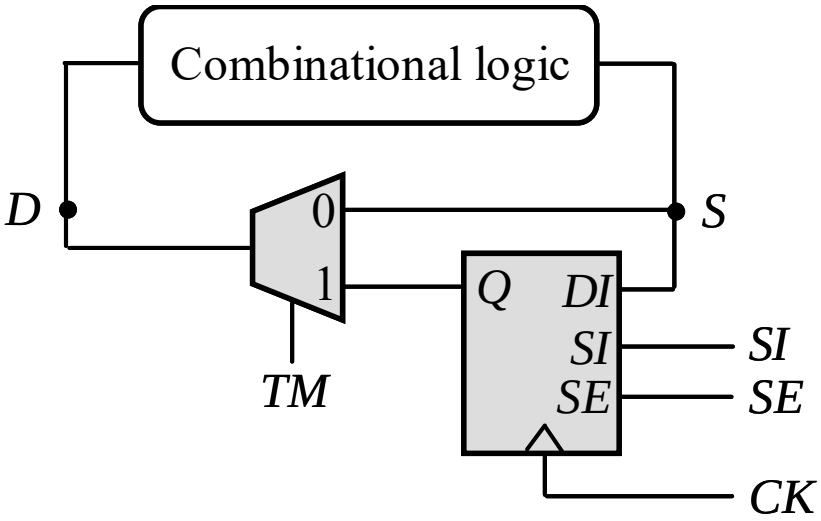
(a) Original circuit

The best way is to rewrite the RTL code.

Depending on whether the number of inversions on a combinational feedback loop is even or odd, it can introduce either sequential behavior or oscillation into a design.

Since the value stored in the loop cannot be controlled or determined during test, this can lead to an increase in test generation complexity or fault coverage loss.

Combinational Feedback Loops

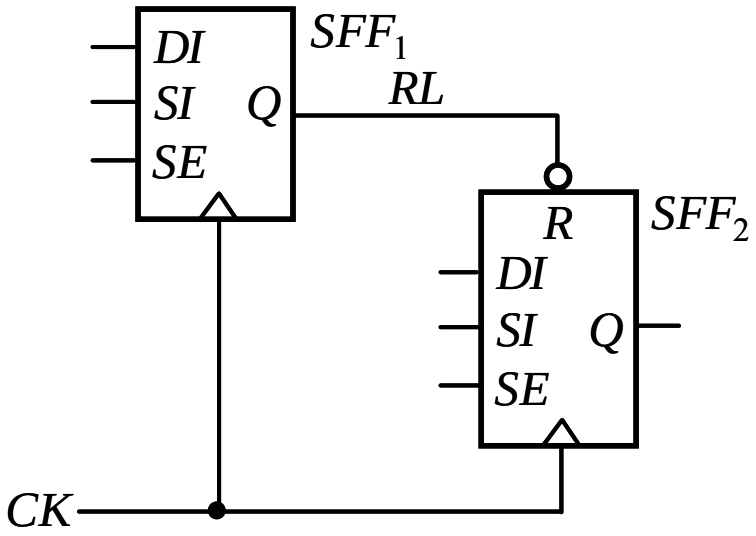


(b) Modified circuit

It can be fixed by using a test mode signal *TM*.

This signal permanently disables the loop throughout the entire shift and capture operations, by inserting a scan point to break the loop.

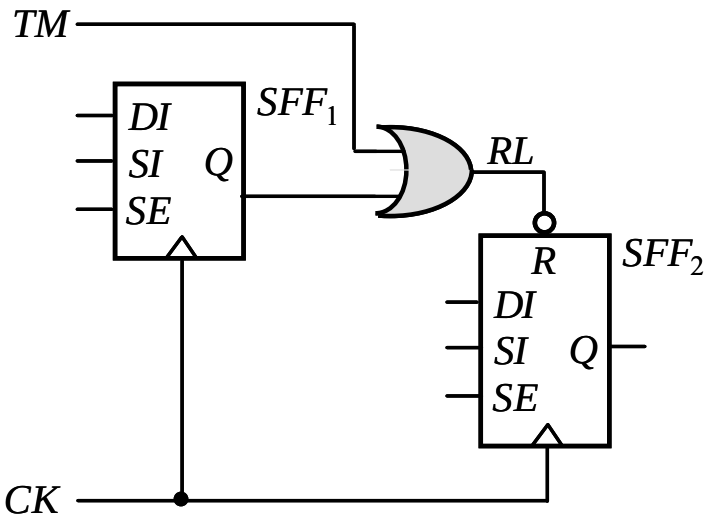
Asynchronous Set/Reset Signals



(a) Original circuit

Asynchronous set/reset signals of scan cells that are not directly controlled from primary inputs can prevent scan chains from shifting data properly.

Asynchronous Set/Reset Signals

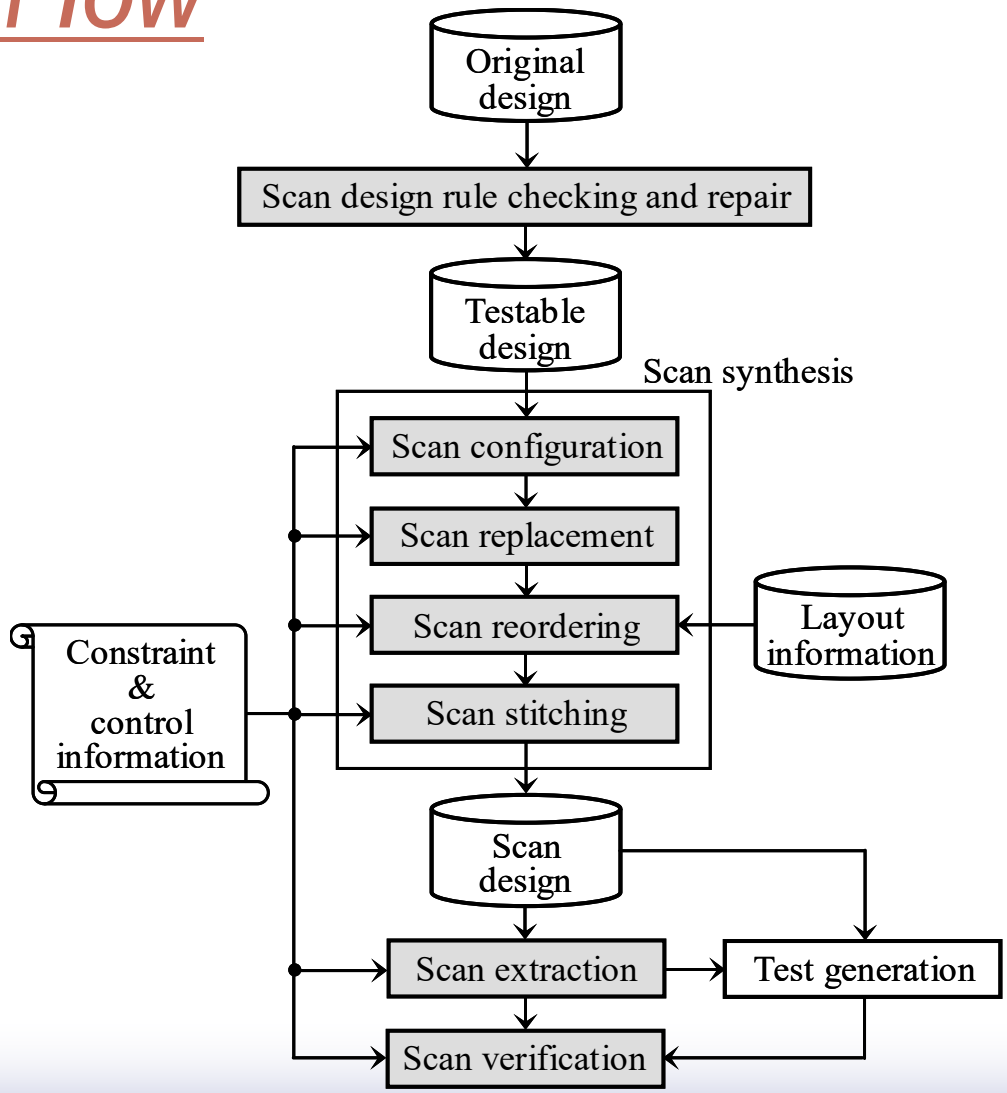


(b) Modified circuit

To avoid this problem, these asynchronous set/reset signals are forced to an inactive state during the shift operation.

Use an OR gate with an input tied to the test mode signal *TM* .When *TM* = 1, the asynchronous reset signal *RL* of scan cell SFF_2 is permanently disabled during the entire test operation.

Scan Design Flow



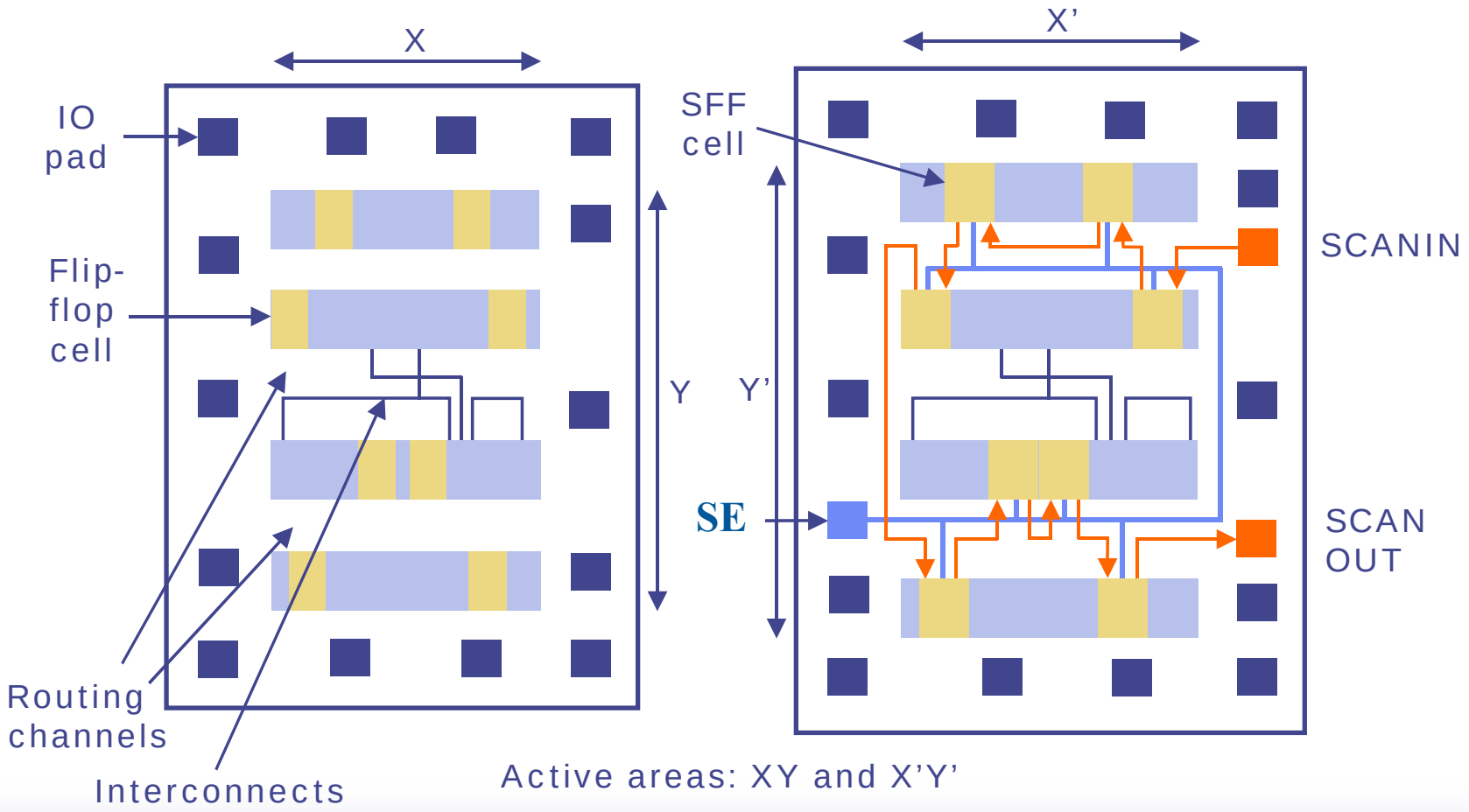
Scan Design Flow

- Scan Design Rule Checking and Repair
  - Identify and repair all scan design rule violations to convert the original design into a testable design
  - Also performed after scan synthesis to confirm that no new violations exist
- Scan Synthesis
  - Converts a testable design into a scan design without affecting the functionality of the original design
    - Scan Configuration
    - Scan Replacement
    - Scan Reordering
    - Scan Stitching

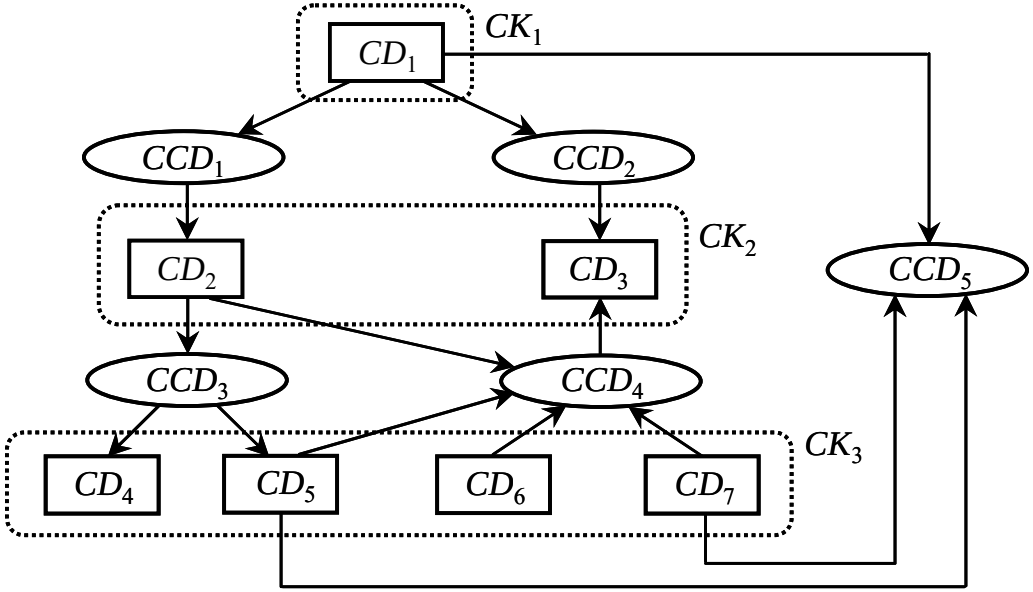
Scan Design Flow

- Scan Extraction
  - Is the process used for extracting all scan cell instances from all scan chains specified in the scan design
- Scan Verification
  - A timing file in *standard delay format* (SDF) which resembles the timing behavior of the manufactured device is used to
    - Verifying the scan shift operation
    - Verifying the scan capture operation
- Scan Design Costs
  - Area overhead cost:
  - I/O pin cost
  - Performance degradation cost
  - Design effort cost

Optimum Scan Layout



Scan Design Rule Checking and Repair



Clock grouping example

An arrow means a data transfers from one clock domain to a different clock domain.

7 clock domains, $CD_1 \sim CD_7$

5 crossing-clock-domain data paths, $CCD_1 \sim CCD_5$

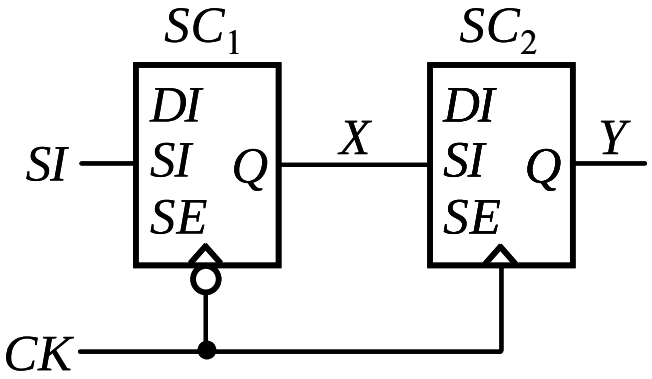
clock skew < *data path delay* + *clock-to-Q delay* (originating clock)

Scan Synthesis

- Includes four separate and distinct steps:
  - Scan Configuration
    - The number of scan chains used: **limit of pins and tester channels**
    - The types of scan cells used to implement these scan chains: **lib**
    - Which storage elements to exclude from the process: **timing, security**
    - How the scan cells are arranged: **clock domains**
  - Scan Replacement
    - Replaces all original storage elements in the testable design with their functionally-equivalent scan cells
  - Scan Reordering
    - The process of reordering the scan chains based on the physical scan cell locations, in order to minimize the amount of interconnect wires used to implement the scan chains
  - Scan Stitching
    - Stitch all scan cells together to form scan chains

Scan Synthesis - Scan Configuration

Mixing negative-edge and positive-edge scan cells in a scan chain

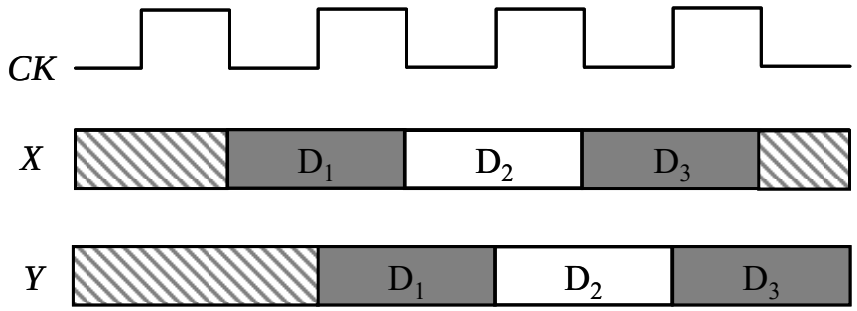


Circuit Structure

This circuit structure comprising a negative-edge scan cell followed by a positive-edge scan cell.

Scan Synthesis - Scan Configuration

Mixing negative-edge and positive-edge scan cells in a scan chain

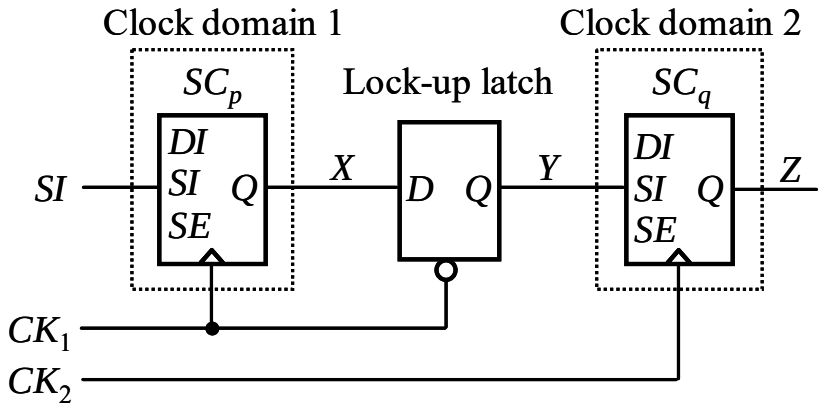


Timing Diagram

Y will first take on the state *X* at the rising *CK* edge, before *X* is loaded with the *SI* value at the falling *CK* edge.

If we accidentally place the positive-edge scan cell before the negative-edge scan cell, both scan cells will always incorrectly contain the same value at the end of each shift clock cycle.

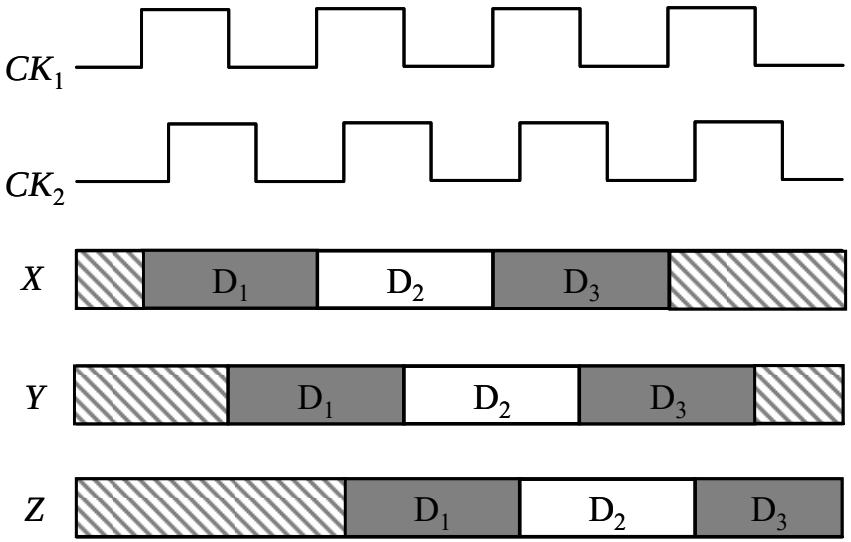
Scan Synthesis - Scan Configuration



Circuit Structure

A lock-up latch is inserted between adjacent cross-clock-domain scan cells, in order to guarantee that any clock skew between the clocks can be tolerated.

Scan Synthesis - Scan Configuration



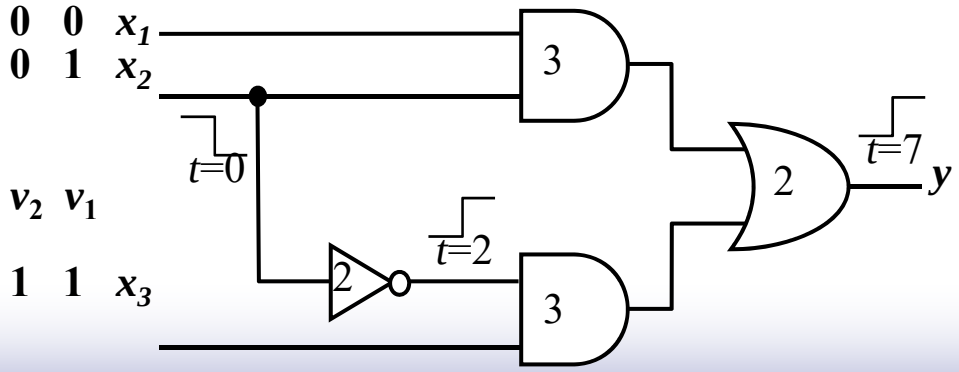
Timing diagram

During each shift clock cycle, *X* will first take on the *SI* value at the rising CK_1 edge. Then, *Z* will take on the *Y* value at the rising CK_2 edge.

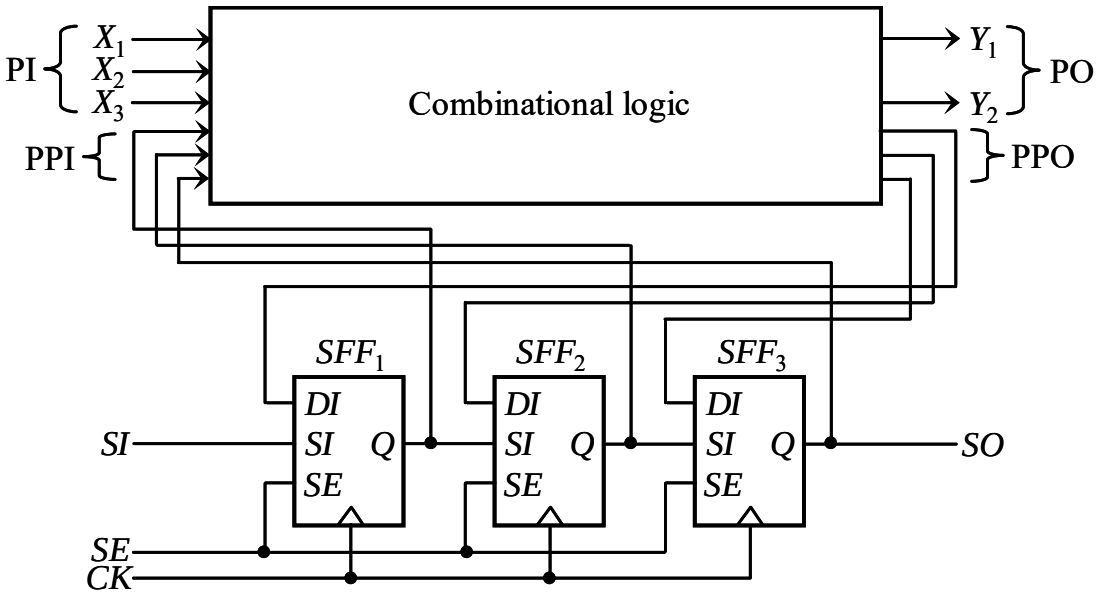
clock skew < width (duty cycle)

Delay Faults and Crosstalk

- Path-delay fault model considers cumulative propagation delay through CUT
  - 2 test vectors create transition along path
  - Faulty circuit has excessive delay
- Delays and glitches can be caused by crosstalk between interconnect
  - due to inductance and capacitive coupling



问题：Normal Scan能否支持2-vector 测试？



Normal Scan (Muxed-D full-scan circuit)

时延故障：典型的需要2-vector (V1, V2) 测试的故障模型

V1用于将电路稳定在需要的状态；
V2用于产生信号的跳变、并将其传播到组合电路的输出

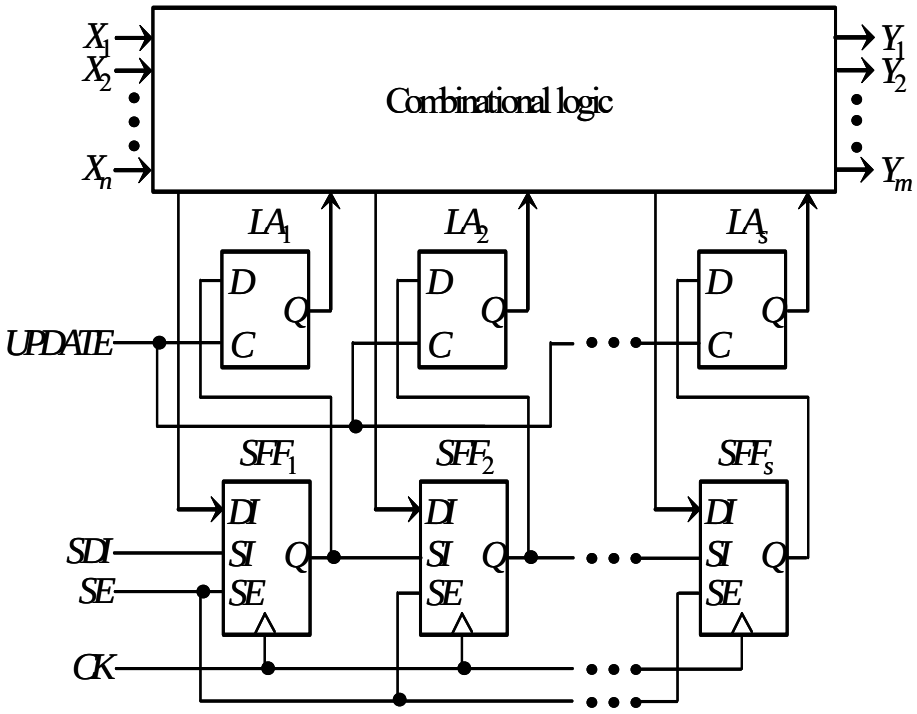
Special-Purpose Scan Designs

- Enhanced scan
- Snapshot scan
- Error-resilient scan

Enhanced Scan

- Why enhanced scan is introduced?
  - Testing for a delay fault requires a pair of test vector at speed
  - Be able to capture the response to the transiton at operating frequency
- What is new?
  - Allow the typical scan cell to store two bits of data
  - Achieved through the addition of a D latch

Enhanced-scan Architecture and Operation



Enhanced-scan architecture

The first test vector V_1 is first shifted into the scan cells ($SFF_1 \sim SFF_s$) and then stored into the additional latches ($LA_1 \sim LA_s$) when the *UPDATE* signal is set to 1.

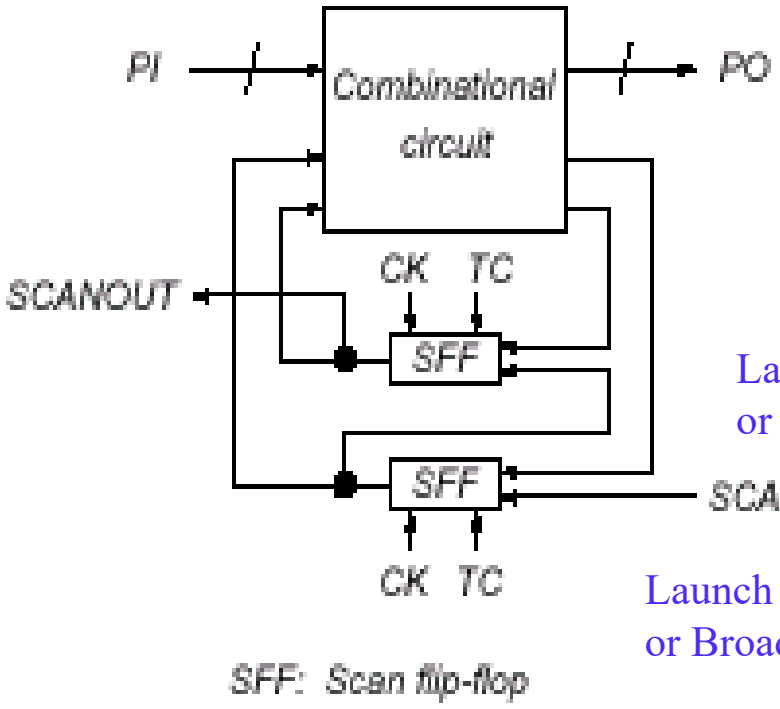
The second test vector V_2 is shifted into the scan cells while the *UPDATE* signal is set to 0.

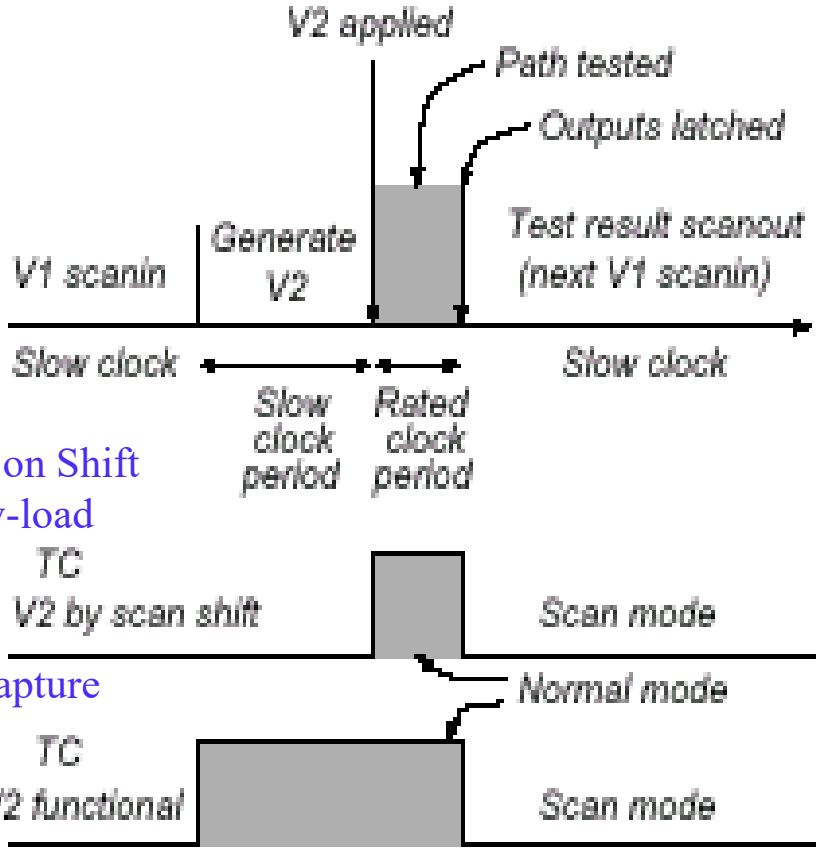
Enhanced Scan - Advantages & Disadvantages

- Advantages
  - High delay fault coverge achieved by applying any arbitrary pair of test vectors
- Disadvantages
  - Requiring an additional scan-hold D latch
  - Difficulty of maintaining the timing relationship between *UPDATE* and *CK*
  - Over-test problem caused by false paths

Normal-Scan Delay Test

- The vector-pairs: especially generated.

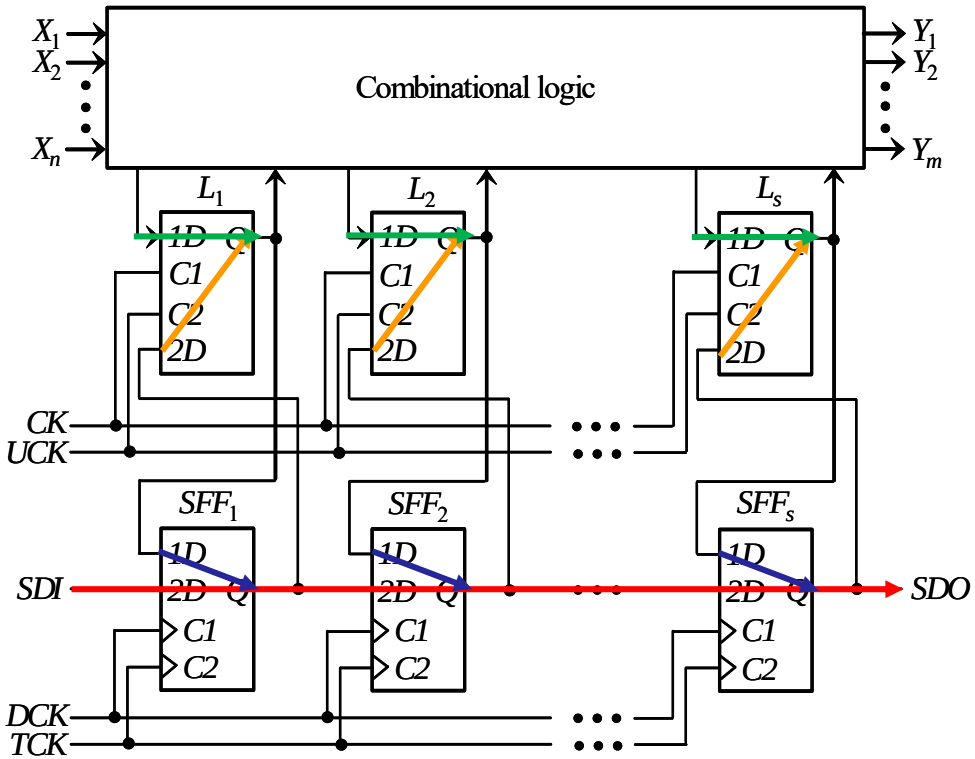




Snapshot scan

- Why snapshot scan is introduced?
  - Capture a snapshot of the internal states
  - Without disruption of the functional operation
- What is new?
  - Add a scan cell (2-port D latches) to each storage element of interest
  - Implement scan-set architecture

Snapshot scan



Scan-set architecture

(1) Test data can be shifted into and out of the scan cells ($SFF_1 \sim SFF_s$) from the *SDI* and *SDO* pins using *TCK*.

(2) The test data can be transferred to the system latches ($L_1 \sim L_s$) in parallel through their *2D* inputs using *UCK*.

(3) The circuit can be operated in normal mode using *CK* to capture the values from the combinational logic into the system latches ($L_1 \sim L_s$).

(4) The system latch contents can be loaded into the scan flip-flops through their *1D* inputs using *DCK*.

Snapshot scan - Advantages & Disadvantage

- Advantages
  - Significantly improve the circuit's diagnostic resolution and silicon debug capability
  - Allow on-chip, on-board and in-system debug and diagnosis
- Disadvantage
  - Increased area overhead

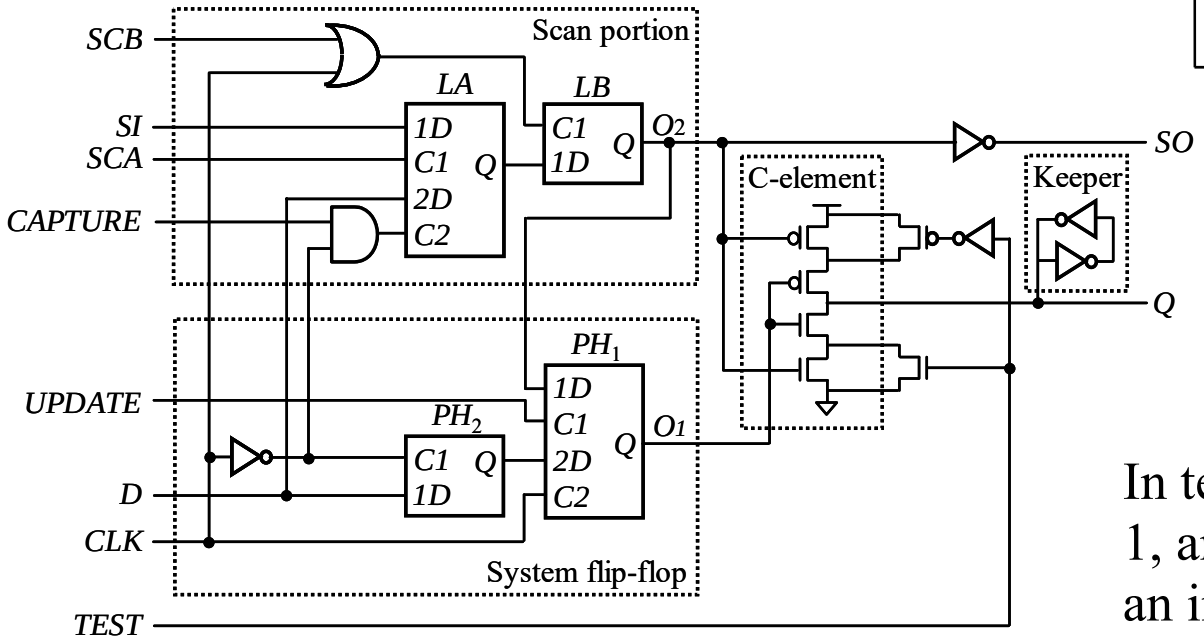
Error-Resilient scan

- Why is error-resilient scan introduced?
  - Soft errors are transient single-event upsets with various causes
  - Soft errors increase as shrinking IC geometry and increasing frequency
  - Reliability concerns are created for protecting a device from soft errors
- What can error-resilient scan do?
  - Observe soft errors occurring in memories and storage elements
  - Observe a transient fault in a combinational gate captured by a memory or storage element

Error-Resilient Scan

C-element truth table

O_1	O_2	Q
0	0	1
1	1	0
0	1	Previous value retained
1	0	Previous value retained



Error-resilient scan cell

In test mode, *TEST* is set to 1, and the C-element acts as an inverter.
In system mode, *TEST* is set to 0, and the C-element acts as a hold-state comparator.

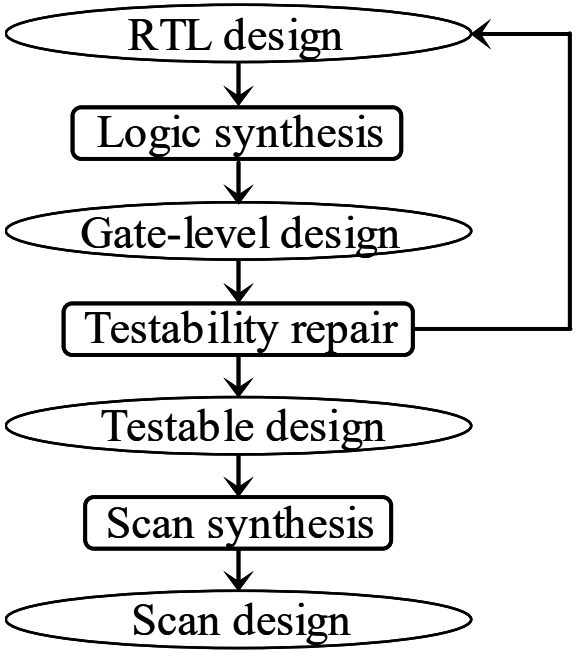
Error-Resilient scan - Advantages & Disadvantages

- Advantages
  - Provide online detection and correction of soft errors
  - Embed with scan testing capability
- Disadvantages
  - Require many test signals and clocks
  - Area overhead

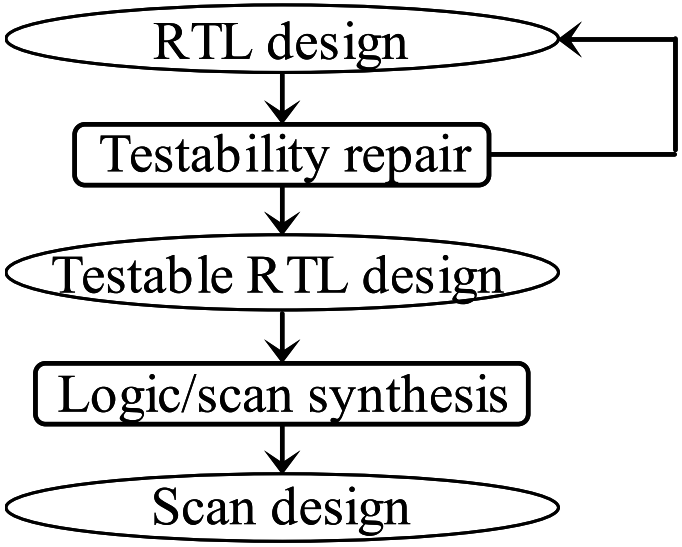
RTL Design for Testability

- Why are RTL designs needed?
  - Growth of device number
  - Tight timing
  - Potential yield loss
  - Low-power issues
  - Increased core reusability
  - Time-to market pressure

Comparison of design flows at RTL and Gate-level



Gate-level testability repair design flow



RTL testability repair design flow

RTL Scan Design Rule Checking

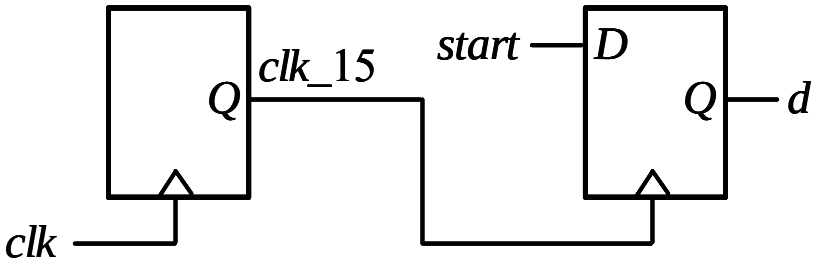
- Fast synthesis
  - Mapped onto combinational primitives and high-level models
- Identify testability problems
  - Static solutions (without simulation)
  - Dynamic solutions (with simulation)

RTL Scan Design Repair – An Example

- original

```
always @(posedge clk)
   if (q == 4'b1111)
      clk_15 <= 1;
   else
      begin
         clk_15 <= 0;
         q <= q + 1;
      end
always @(posedge clk_15)
   d < = start;
```

(a) Generated clock (RTL code)



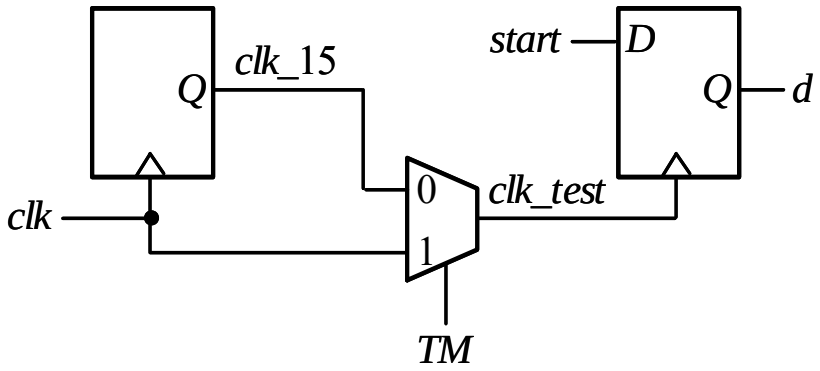
(b) Generated clock (Schematic)

RTL Scan Design Repair – An Example

- Atuomatic repair at the RTL using *TM*

```
always @(posedge clk)
   if (q == 4'b1111)
      clk_15 <= 1;
   else
      begin
         clk_15 <= 0;
         q <= q + 1;
      end
assign clk_test = (TM)? clk : clk_15;
always @(posedge clk_test)
   d <= start;
```

(c) Generated clock (RTL code)



(d) Generated clock repair (Schematic)

RTL Scan Synthesis

- RTL scan synthesis
  - The scan equivalent of each storage element refers to an RTL structure
  - The scan chains are inserted into the RTL design
- Pseudo RTL scan synthesis
  - Specify pseudo primary inputs and pseudo primary outputs
  - Can cope with many other DFT structures
  - Perform one-pass or single-pass synthesis

RTL Scan Extraction and Verification

- Scan extraction
  - Rely on performing fast synthesis on the RTL scan design
  - Generate a software model for tracing the scan connection
- Scan verification
  - Rely on generating a flush testbench to simulate flush tests
  - The flush testbench can be used for both RTL and gate-level designs
  - Apply broadside-load test for verifying the scan capture operation at RTL

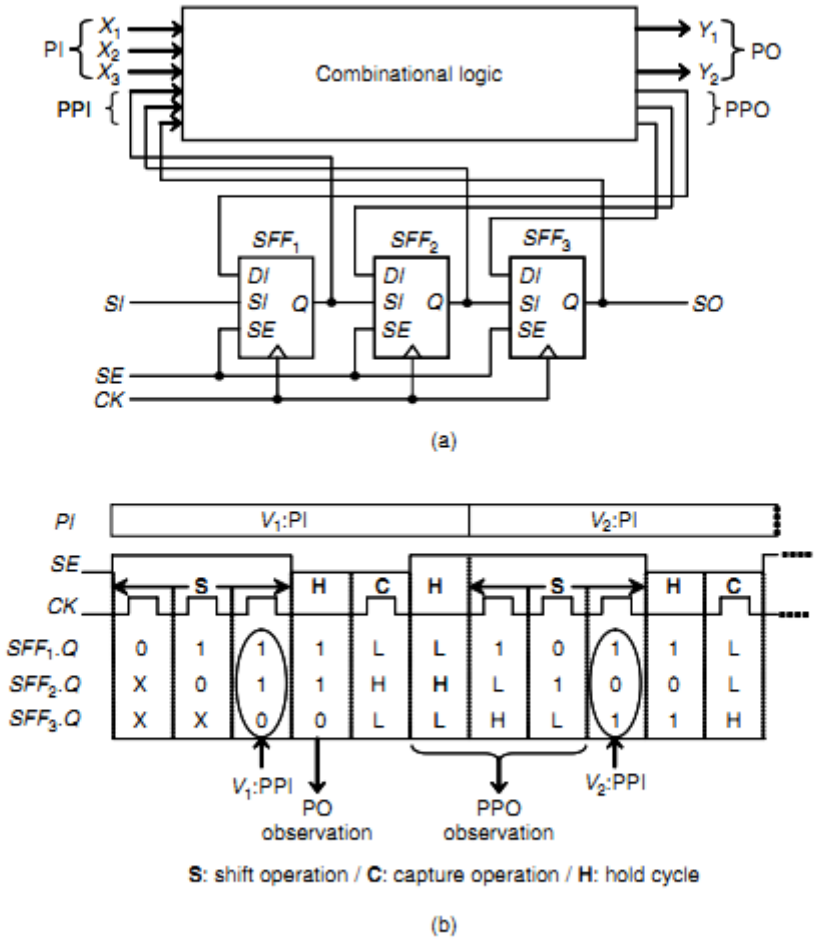
Concluding Remarks

- DFT has become vital for ensuring product quality
- Scan design is the most widely used DFT technique
- New design and test challenges
  - Further reduce test power, test data volume and test application time
  - Cope with physical failures of the nanometer design era

Exercises

- **2.9 (Full-Scan Design) Calculate the number of clock cycles required for testing a full-scan design with n test vectors. Assume that the full-scan design has m scan chains, each having the same length L, and that scan testing is conducted in the way shown in Figure 2.14b.**

Sequential circuit example.



■ **FIGURE 2.14**

Muxed-D full-scan circuit and test operations: (a) muxed-D full-scan circuit, and (b) test operations.

Exercises

- **2.17 (Test Signal) Describe the difference between the test mode signal TM and the scan enable signal SE used in scan testing.**

- **2.19 (RTL Testability Enhancement) Read the following Verilog HDL code and draw its schematic. Then determine if there is any scan design rule violation. If there is any violation, modify the RTL code to fix the problem, then draw the schematic of the modified RTL code.**

```
reg [3:0] tri_en;
always @(posedge clk)
begin
case (bus_sel)
0: tri_en[0] = 1'bl;
1: tri_en[1] = 1'bl;
2: tri_en[2] = 1'bl;
3: tri_en[3] = 1'bl;
endcase
end
assign dbus = (tri_en[0])? d1 : 8'bz;
assign dbus = (tri_en[1])? d2 : 8'bz;
assign dbus = (tri_en[2])? d3 : 8'bz;
assign dbus = (tri_en[3])? d4 : 8'bz;
```

下次课预告

时间：2021年09月29日（周三6:10pm）

地点：教1-109

内容：逻辑与故障模拟

教材：VLSI TEST PRINCIPLES AND ARCHITECTURES

Chapter 3 Logic and Fault Simulation

教学安排：13次课

课次	内容	教材	教师	日期
第1讲	VLSI测试技术导论	第1章	李晓维	2021-09-08
第2讲	可测试性设计	第2章	李晓维	2021-09-15
第3讲	可测试性设计	第2章	李晓维	2021-09-22
第4讲	逻辑与故障模拟	第3章	李华伟	2021-09-29
第5讲	测试生成	第4章	李华伟	2021-10-06
第6讲	逻辑自测试	第5章	李华伟	2021-10-20
第7讲	测试压缩	第6章	韩银和	2021-11-03
第8讲	逻辑诊断	第7章	叶靖/李晓维	2021-11-10
第9讲	存储器测试与BIST	第8章	韩银和	2021-11-17
第10讲	存储器诊断与BISR	第9章	韩银和	2021-11-24
第11讲	边界扫描与SOC测试	第10章	李华伟	2021-12-01
第12讲	纳米电路测试技术	第12章	李华伟	2021-12-08
第13讲	总结与复习		李华伟	2021-12-15
考试	课堂开卷		李晓维	2021-12-22