國立清華大學電機系

EE-6250
超大型積體電路測試
VLSI Testing

Chapter 5
Design For Testability
& Scan Test

Outline

- Introduction
  - Why DFT?
  - What is DFT?
- Ad-Hoc Approaches
- Full Scan
- Partial Scan

Why DFT ?

- Direct Testing is Way Too Difficult !
  - Large number of FFs
  - Embedded memory blocks
  - Embedded analog blocks
- Design For Testability is inevitable
  - Like death and tax

Design For Testability

- Definition
  - Design For Testability (DFT) refers to those design techniques that make test generation and testing cost-effective
- DFT Methods
  - Ad-hoc methods
  - Scan, full and partial
  - Built-In Self-Test (BIST)
  - Boundary scan
- Cost of DFT
  - Pin count, area, performance, design-time, test-time

Why DFT Isn't Universally Used Previously?

- Short-sighted view of management
- Time-to-market pressure
- Life-cycle cost ignored by development management/contractors/buyers
- Area/functionality/performance myths
- Lack of knowledge by design engineers
- Testing is someone else's problem
- Lack of tools to support DFT until recently

We don't' have to worry about this management barrier any more
➔ Most design teams now have DfT people

Important Factors

- Controllability
  - Measure the ease of controlling a line
- Observability
  - Measure the ease of observing a line at PO
- Predictability
  - Measure the ease of predicting output values
- DFT deals with ways of improving
  - Controllability
  - Observability
  - Predictability

Outline

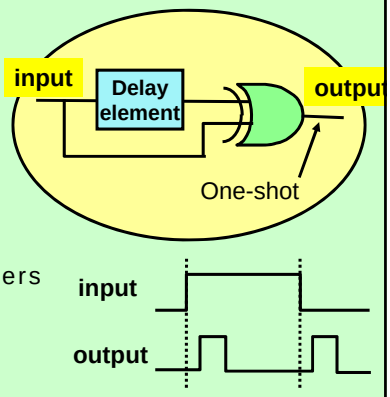
- Introduction
- Ad-Hoc Approaches
  - Test Points
  - Design Rules
- Full Scan
- Partial Scan

Ad-Hoc Design For Testability

- Design Guidelines
  - Avoid redundancy
  - Avoid asynchronous logic
  - Avoid clock gating (e.g., ripple counter)
  - Avoid large fan-in
  - Consider tester requirements (tri-stating, etc.)
- Disadvantages
  - High fault coverage not guaranteed
  - Manual test generation
  - Design iterations required

Some Ad-Hoc DFT Techniques

- Test Points
- Initialization
- Monostable multivibrators
  - One-shot circuit
- Oscillators and clocks
- Counters / Shift-Registers
  - Add control points to long counters
- Partition large circuits
- Logical redundancy
- Break global feedback paths



ch5-9

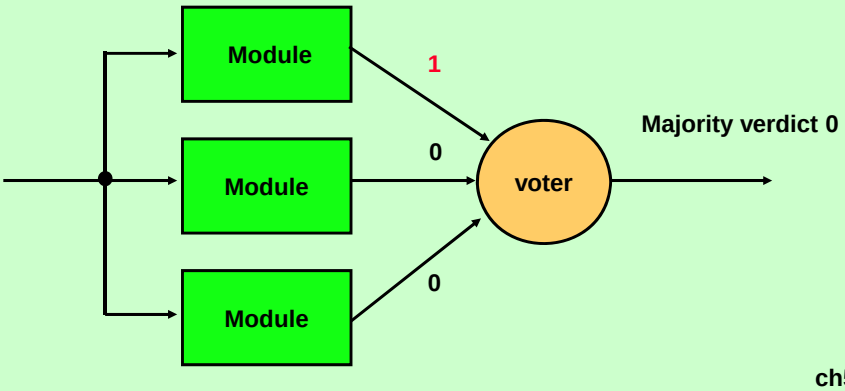
On-Line Self-Test & Fault Tolerance By Redundancy

- Information Redundancy
  - Outputs = (information-bits) + (check-bits)
  - Information bits are the original normal outputs
  - Check bits always maintains a specific pre-defined logical or mathematical relationship with the corresponding information bits
  - Any time, if the information-bits and check-bits violate the pre-defined relationship, then it indicates an error
- Hardware Redundancy
  - Use extra hardware (e.g., duplicate or triplicate the system) so that the fault within one module will be masked (I.e., the faulty effect never observed at the final output)

ch5-10

Module Level Redundancy

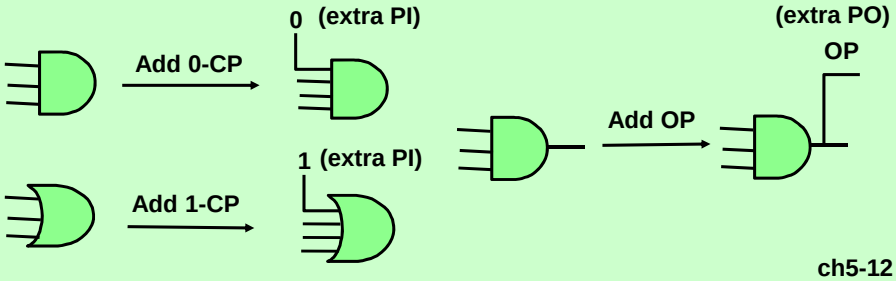
- Triple Module Redundancy (TMR)
  - majority voting on three identical modules' outputs help mask out faults that occur in a single module



ch5-11

Test Point Insertion

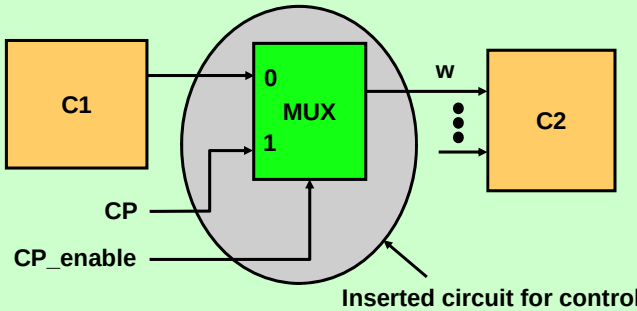
- **Employ test points to enhance**
  - **Controllability**
  - **Observability**
- **CP: Control Points**
  - **Primary inputs used to enhance controllability**
- **OP: Observability Points**
  - **Primary outputs used to enhance observability**



ch5-12

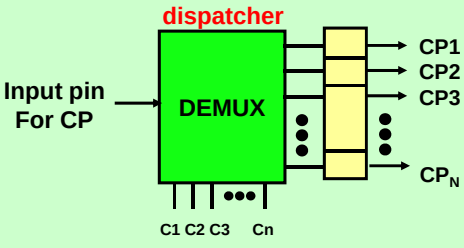
0/1 Injection Circuitry

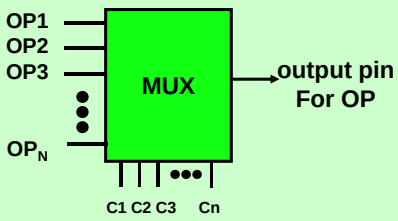
- Normal operation
  When CP_enable = 0
- Inject 0
  - Set CP_enable = 1 and CP = 0
- Inject 1
  - Set CP_enable = 1 and CP = 1



Single I/O Port for Multiple Test Points

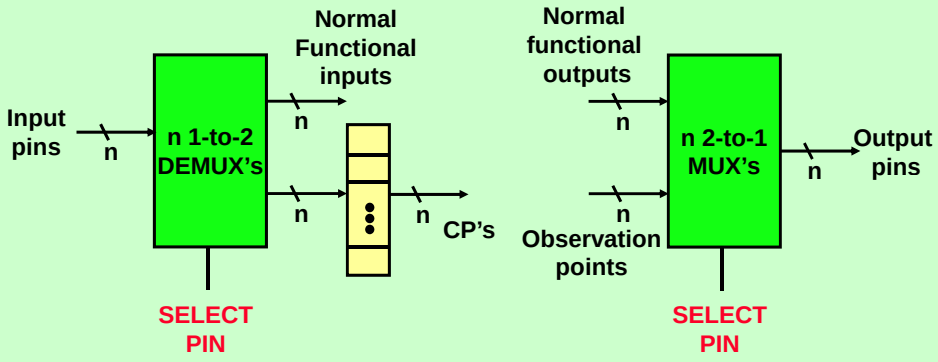
- Constraints of using test points
  - A large demand on I/O pins
  - This constraint can be somewhat relieved by using MUX & DEMUX at the cost of increasing the test time





Sharing Between Test Points & Normal I/O

- Advantage: Even fewer I/O pins for Test Points
- Overhead: Extra MUX delay for normal I/O

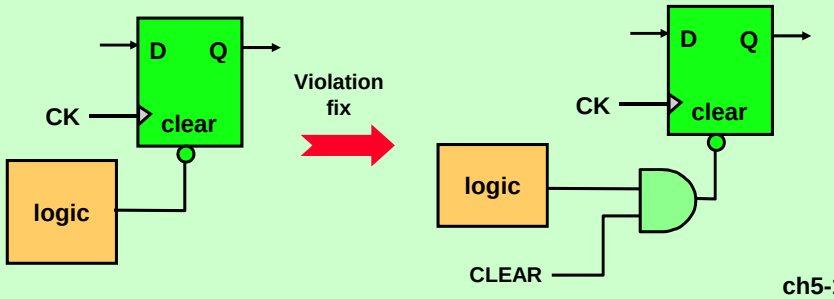


Control Point Selection

- Impact
  - The controllability of the fanout-cone of the added point is improved
- Common selections
  - Control, address, and data buses
  - Enable / Hold inputs
  - Enable and read/write inputs to memory
  - Clock and set/clear signals of flip-flops
  - Data select inputs to multiplexers and demultiplexers

Example: Use CP to Fix DFT Rule Violation

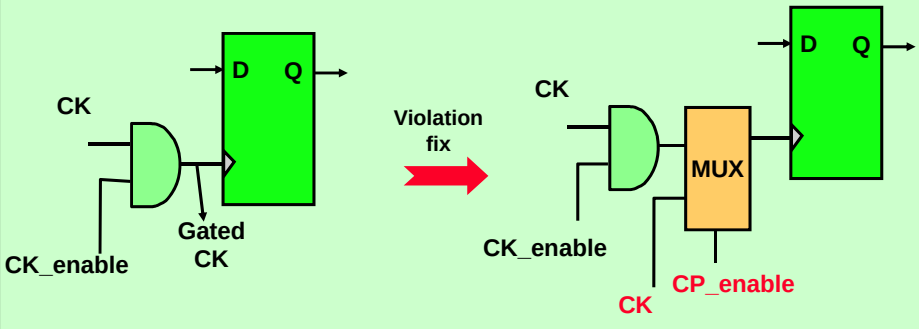
- DFT rule violations
  - The set/clear signal of a flip-flop is generated by other logic, instead of directly controlled by an input pin
  - Gated clock signals
- Violation Fix
  - Add a control point to the set/clear signal or clock signals



ch5-17

Example: Fixing Gated Clock

- Gated Clocks
  - Advantage: power dissipation of a logic design can thus reduced
  - Drawback: the design's testability is also reduced
- Testability Fix

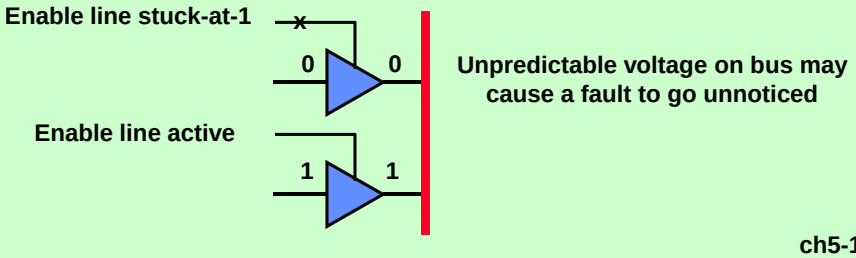


ch5-18

Example: Fixing Tri-State Bus Contention

- **Bus Contention**
  - A stuck-at-fault at the tri-state enable line may cause bus contention – multiple active drivers are connected to the bus simultaneously
- **Fix**
  - Add CPs to turn off tri-state devices during testing

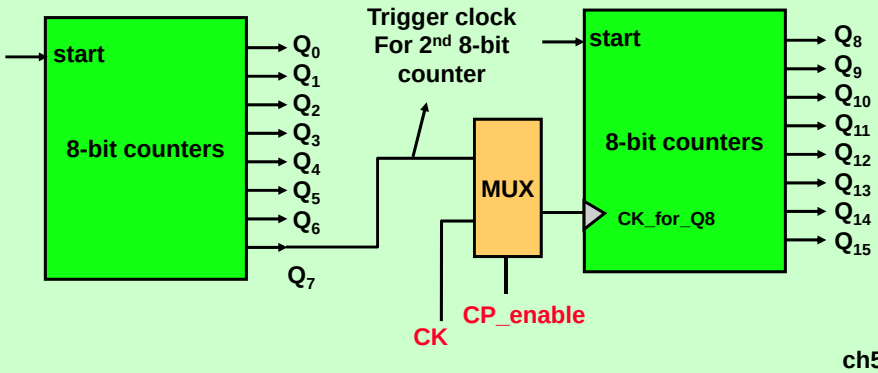
(A Bus Contention Scenario in the presence of a fault)



ch5-19

Example: Partitioning Counters

- **Consider a 16-bit ripple-counter**
  - Could take up to $2^{16} = 65536$ cycles to test
  - After being partitioned into two 8-bit counters below, it can be tested with just $2x2^8 = 512$ cycles



ch5-20

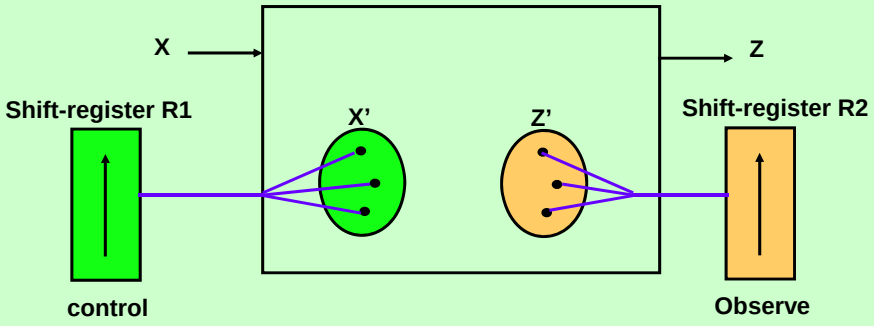
Observation Point Selection

- Impact
  - The observability of the fanin-cone (or transitive fanins) of the added OP is improved
- Common choice
  - Stem lines having a large number of fanouts
  - Global feedback paths
  - Redundant signal lines
  - Output of logic devices having many inputs
    - MUX, XOR trees
  - Output from state devices
  - Address, control and data buses

(常為電路區塊間之介面訊號)

Problems of CP & OP

- **Large number of I/O pins**
  - **Add MUXes to reduce the number of I/O pins**
  - **Serially shift CP values by shift-registers**
- **Larger test time**



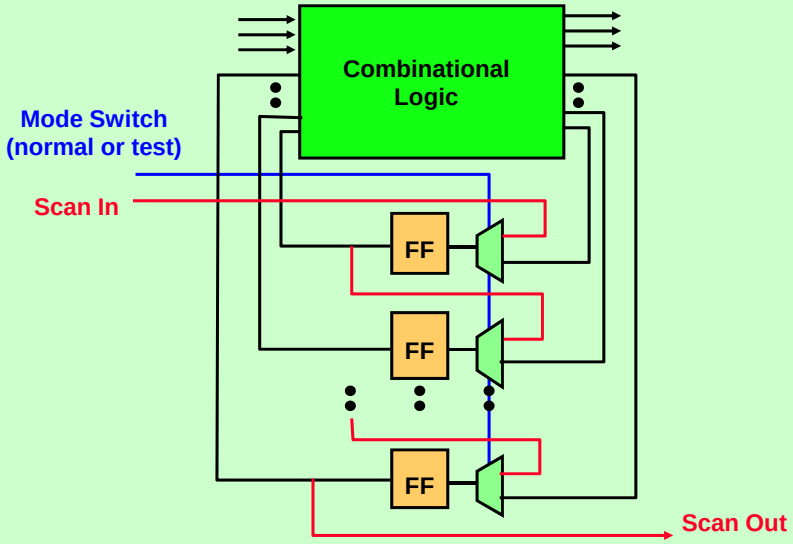
Outline

- Introduction
- Ad-Hoc Approaches
- Full Scan
  - The Concept
  - Scan Cell Design
  - Random Access Scan
- Partial Scan

What Is Scan ?

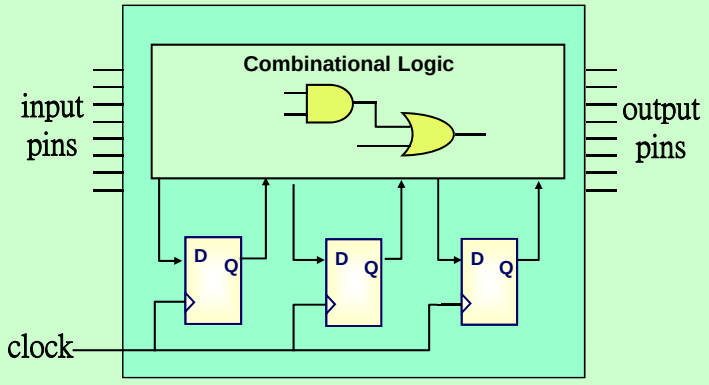
- **Objective**
  - To provide controllability and observability at internal state variables for testing
- **Method**
  - Add test mode control signal(s) to circuit
  - Connect flip-flops to form shift registers in test mode
  - Make inputs/outputs of the flip-flops in the shift register controllable and observable
- **Types**
  - Internal scan
    - Full scan, Partial scan, Random access
  - Boundary scan

The Scan Concept



ch5-25

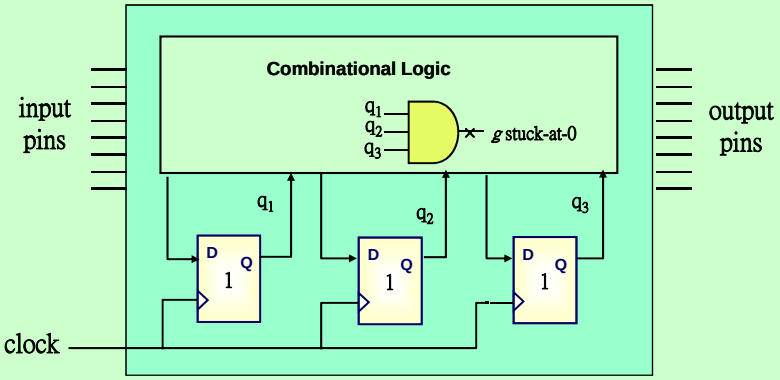
A Logic Design Before Scan Insertion



Sequential ATPG is extremely difficult:
due to the lack of controllability and observability at flip-flops.

ch5-26

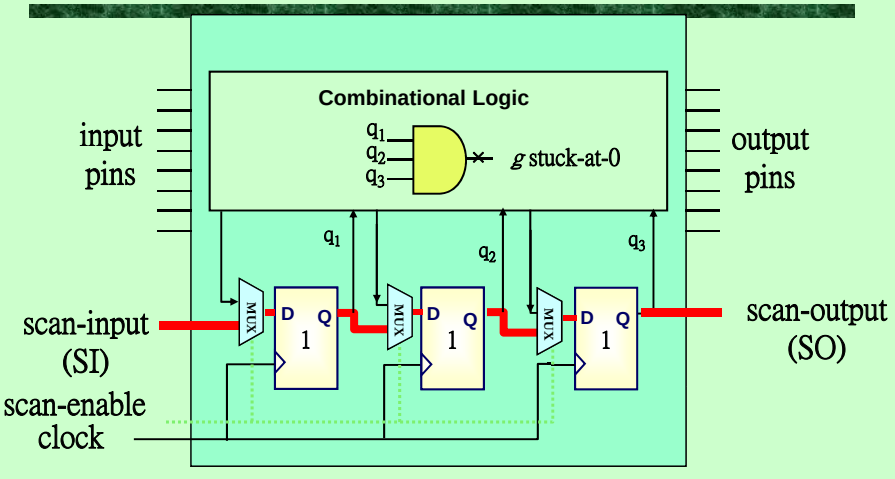
Example: A 3-stage Counter



It takes 8 clock cycles to set the flip-flops to be (1, 1, 1), for detecting the *g* stuck-at-0 fault
(2^{20} clock cycles for a 20-stage counter !)

ch5-27

A Logic Design After Scan Insertion



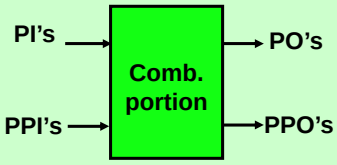
Scan Chain provides an easy access to flip-flops
⟶ Pattern Generation is much easier !!

Note: Scan Enable (SE), not shown here, controls every MUX.

ch5-28

Procedure of Applying Test Patterns

- **Notation**
  - Test vectors $T = < t_i^I, t_i^F > i= 1, 2, \ldots$
  - Output Response $R = < r_i^O, r_i^F > i= 1, 2, \ldots$



- **Test Application**
  - (1) i = 1;
  - (2) Scan-in t_1^F /* scan-in the first state vector for PPI's */
  - (3) Apply t_i^I /* apply current input vector at PI's */
  - (4) Observe r_i^O /* observe current output response at PO's */
  - (5) Capture PPOs to FFs as r_i^F /* capture the response at PPO's to FFs */
    - (Set to 'Normal Mode' by raising SE to '1' for one clock cycle)
  - (6) Scan-out r_i^F while scanning-in t_{i+1}^F /* overlap scan-in and scan-out */
  - (7) i = i+1; Goto step (3)

ch5-29

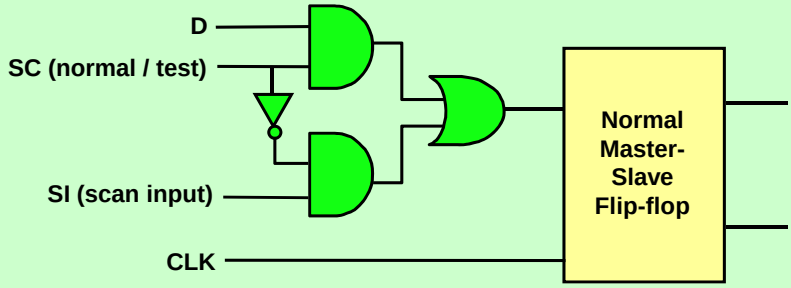
Testing Scan Chain ?

- **Common practice**
  - Scan chain is often first tested before testing the core logic by a so-called flush test - which pumps random vectors in and out of the scan chain
- **Procedure (flush test of scan chain)**
  - (1) i = 0;
  - (2) Scan-in 1st random vector to flip-flops
  - (3) Scan-out (i)th random vector while scanning-in (i+1)th vector for flip-flops.
    - The (i)th scan-out vector should be identical to (i)th vector scanned in earlier, otherwise scan-chain is mal-functioning
  - (4) If necessary i = i+1, goto step (3)

ch5-30

MUX-Scan Flip-Flop

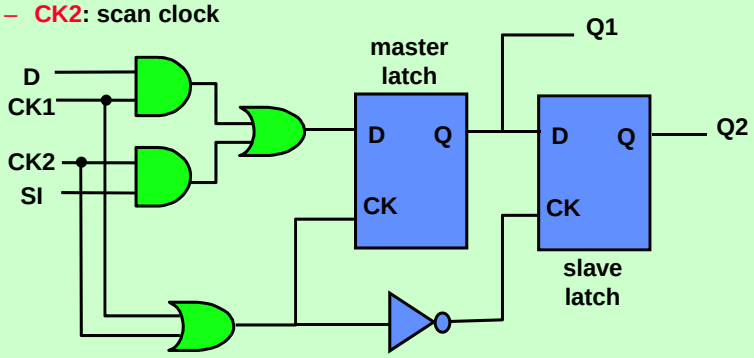
- Only D-type master-slave flip-flops are used
- All flip-flop clocks controlled from primary inputs
  - No gated clock allowed
- Clocks must not feed data inputs of flip-flops
- Most popularly supported in standard cell libraries



ch5-31

Two-Port Dual-Clock Scan FF

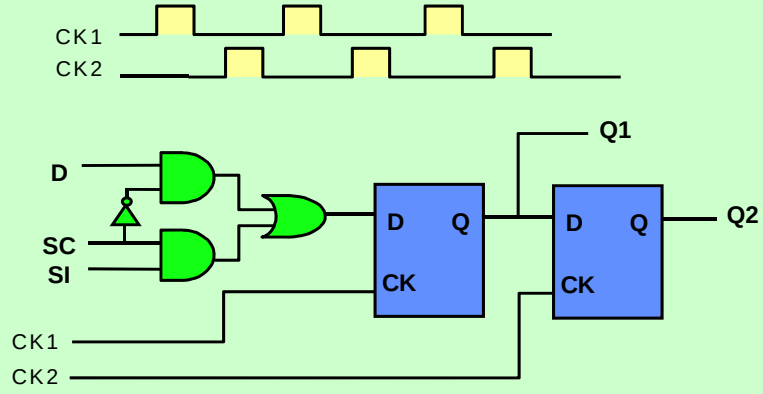
- **Separate normal clock from the clock used for scanning**
  - D: normal input data
  - CK1: normal clock
  - SI: scan input
  - CK2: scan clock



ch5-32

Race-Free Scan FF

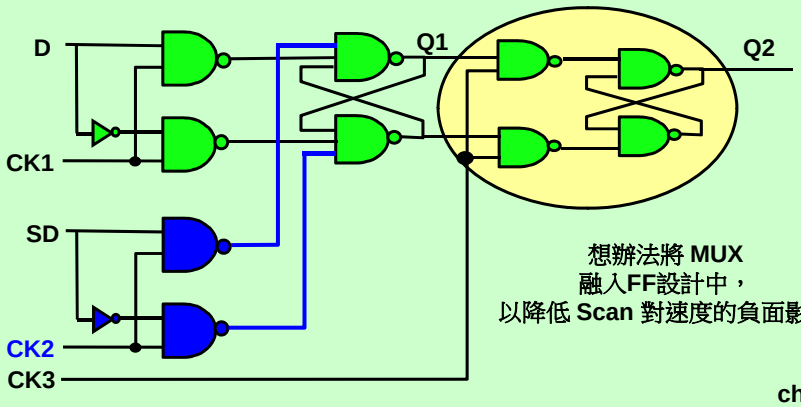
- Use two-phase clocking
  - CK1 and CK2 are two-phase non-overlapping clocks which insure race-free operation



ch5-33

LSSD flip-flop (1977 IBM)

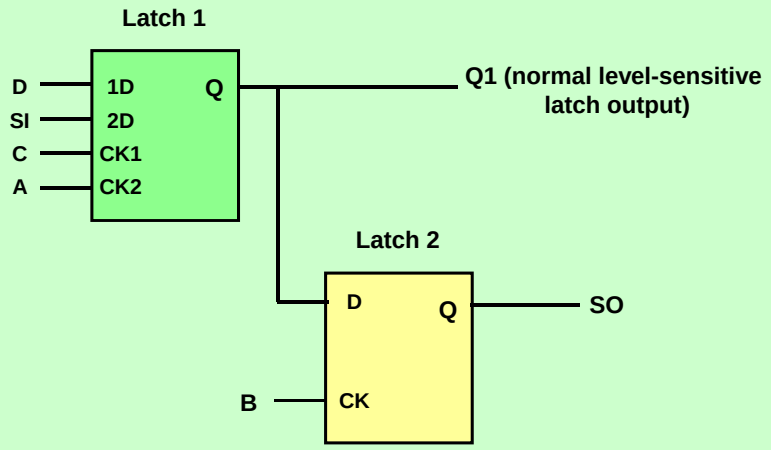
- LSSD: Level Sensitive Scan Design
  - Less performance degradation than MUX-scan FF
- Clocking
  - Normal operation: non-overlapping CK1=1 → CK3=1
  - Scan operation: non-overlapping CK2=1 → CK3=1



想辦法將 MUX
融入FF設計中，
以降低 Scan 對速度的負面影響

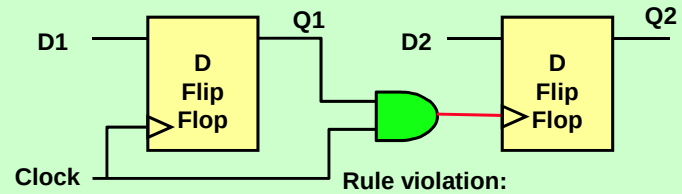
ch5-34

Symbol of LSSD FF



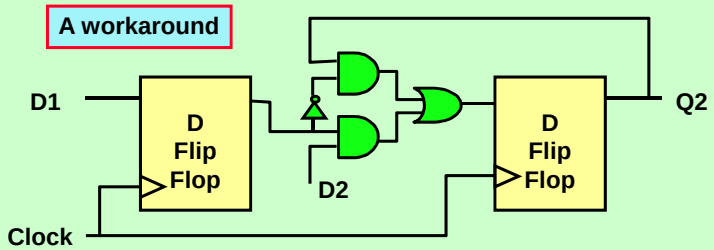
ch5-35

Scan Rule Violation Example



Rule violation:
Flip-flops cannot form a shift-register

A workaround



All FFs are triggered by the same clock edge
Set and reset signals are not controlled by any internal signals

ch5-36

Some Problems With Full Scan

Major Commercial Test Tool Companies
Synopsys
Mentor-Graphics
SynTest (華騰科技)
Cadence

- Problems
  - Area overhead
  - Possible performance degradation
  - High test application time
  - Power dissipation
- Features of Commercial Tools
  - Scan-rule violation check (e.g., DFT rule check)
  - Scan insertion (convert a FF to its scan version)
  - ATPG (both combinational and sequential)
  - Scan chain reordering after layout

ch5-37

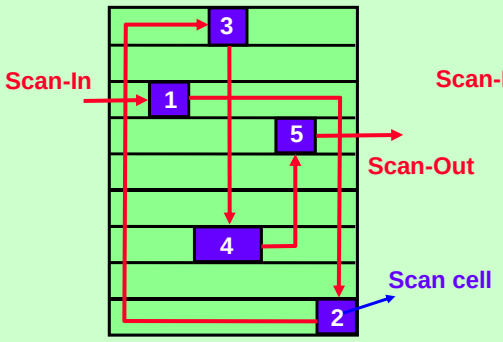
Performance Overhead

- The increase of delay along the normal data paths include:
  - Extra gate delay due to the multiplexer
  - Extra delay due to the capacitive loading of the scan-wiring at each flip-flop's output
- Timing-driven partial scan
  - Try to avoid scan flip-flops that belong to the timing critical paths
  - The flip-flop selection algorithm for partial scan can take this into consideration to reduce the timing impact of scan to the design

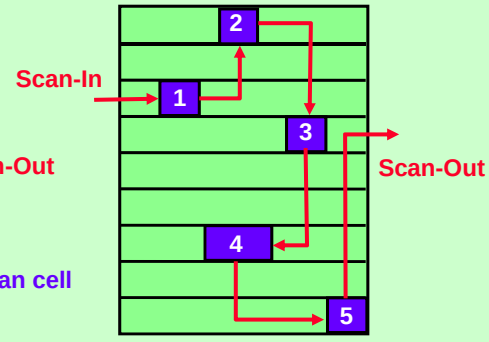
ch5-38

Scan-Chain Reordering

- Scan-chain order is often decided at gate-level without knowing the physical locations of the cells
- Scan-chain consumes a lot of routing resources, and could be minimized by re-ordering the flip-flops in the chain after layout is known



Layout of a scan design



A better scan-chain order

ch5-39

Overhead of Scan Design

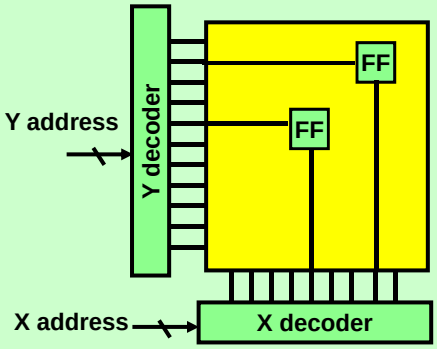
- Number of CMOS gates = 2000
- Fraction of flip-flops = 0.478

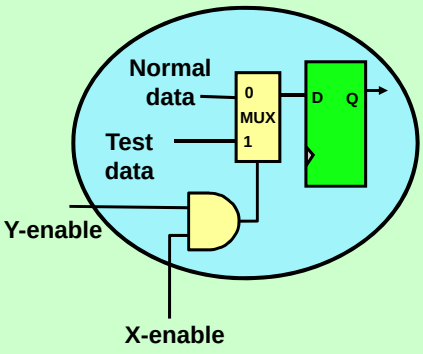
Scan implementation	Predicted overhead	Actual area overhead	Normalized operating frequency
None	0	0	1.0
Hierarchical	14.05%	16.93%	0.87
Optimized	14.05%	11.9%	0.91

ch5-40

Random Access Scan

- Comparison with Scan-Chain
  - More flexible – any FF can be accessed in constant time
  - Test time could be reduced
  - More hardware and routing overhead





ch5-41

Outline

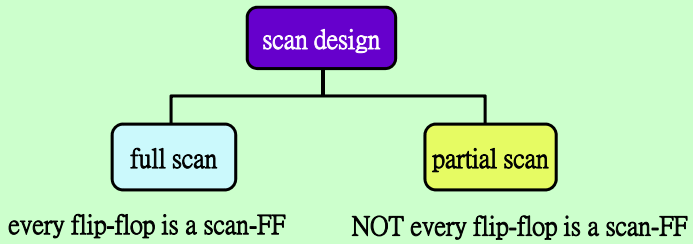
- Introduction
- Ad-Hoc Approaches
- Full Scan
- Partial Scan

ch5-42

Partial Scan

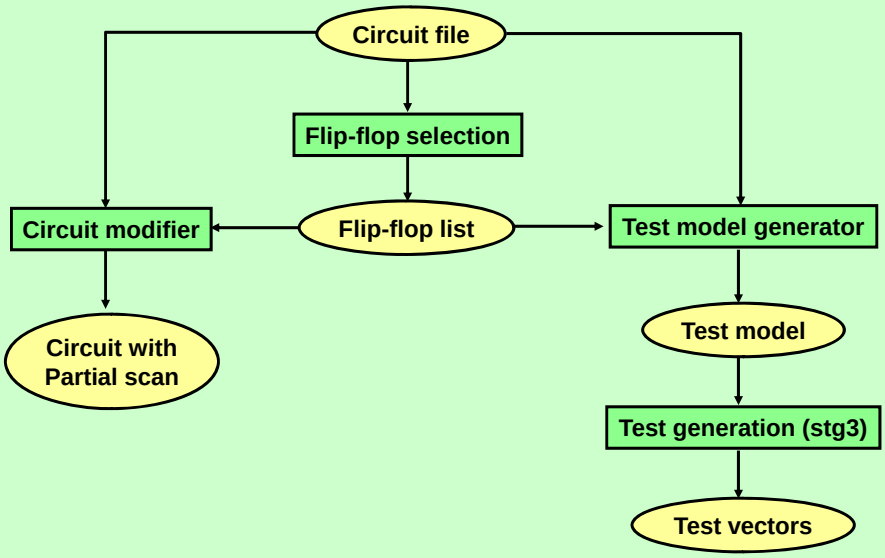
- Basic idea
  - Select a subset of flip-flops for scan
  - Lower overhead (area and speed)
  - Relaxed design rules
- Cycle-breaking technique
  - Cheng & Agrawal, IEEE Trans. On Computers, April 1990
  - Select scan flip-flops to simplify sequential ATPG
  - Overhead is about 25% off than full scan
- Timing-driven partial scan
  - Jou & Cheng, ICCAD, Nov. 1991
  - Allow optimization of area, timing, and testability simultaneously

Full Scan vs. Partial Scan



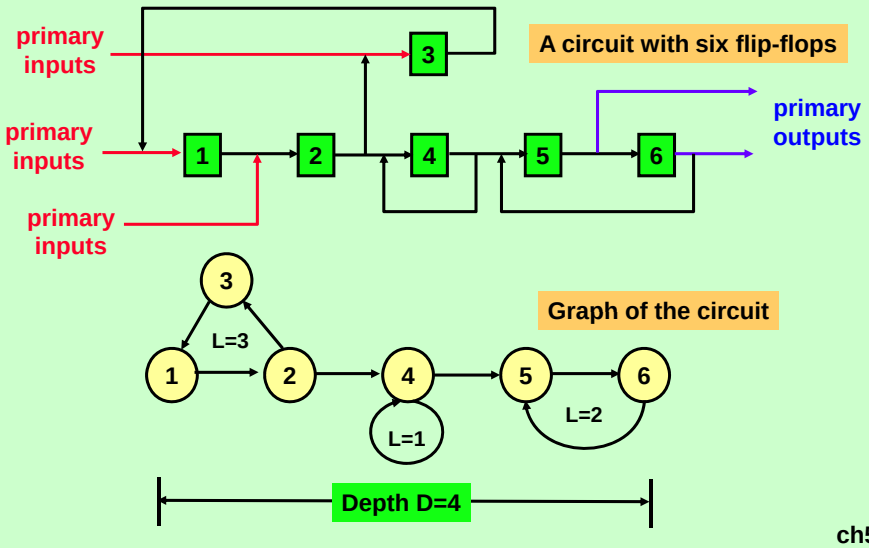
test time	longer	shorter
hardware overhead	more	less
fault coverage	~100%	unpredictable
ease-of-use	easier	harder

A Partial-Scan DfT Flow



ch5-45

Directed Graph Of A Synchronous Sequential Circuit



ch5-46

Partial Scan For Cycle-Free Structure

- Select minimal set of flip-flops
  - To eliminate some or all cycles
- Self-loops of unit length
  - Are not broken to reduce scan overhead
  - The number of self-loops in real design can be quite large
- Limit the length of
  - Consecutive self-loop paths
  - Long consecutive self-loop paths in large circuits may pose problems to sequential ATPG

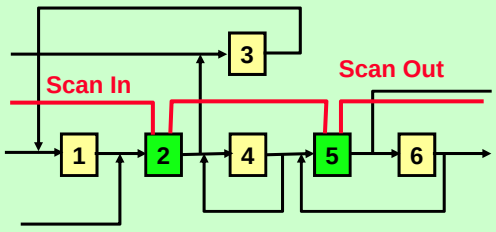
ch5-47

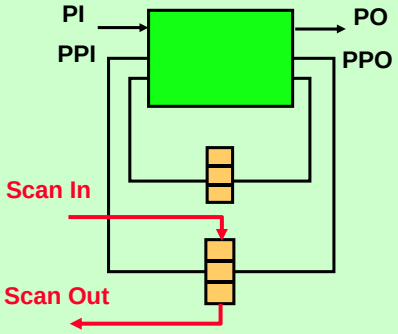
Test Generation for Partial Scan Circuits

- Separate scan clock is used
- Scan flip-flops are removed
  - And their input and output signals are added to the PO/PI lists
- A sequential circuit test generator
  - is used for test generation
- The vector sequences
  - Are then converted into scan sequences
  - Each vector is preceded by a scan-in sequence to set the states of scanned flip-flops
  - A scan-out sequence is added at the end of applying each vector

ch5-48

Partial Scan Design

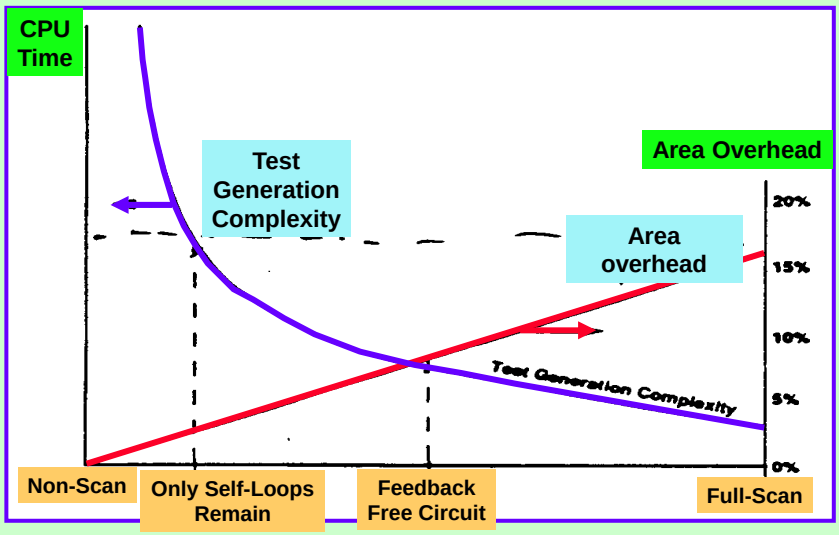




Scan Flip-Flops: {2, 5}
Non-Scan FFs: {1, 3, 4, 6}

ch5-49

Trade-Off of Area Overhead v.s. Test Generation Effort



ch5-50

Summary of Partial-Scan

- Partial Scan
  - Allows the trade-off between test generation effort and hardware overhead to be automatically explored
- Breaking Cycles
  - Dramatically simplifies the sequential ATPG
- Limiting the Length of Self-Loop Paths
  - Is crucial in reducing test generation effort for large circuits
- Performance Degradation
  - Can be minimized by using timing analysis data for flip-flop selection