Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

Preizkušanje elektronskih vezij

Modeliranje napak
Fault modeling

Introduction

- Physical implementation of VLSI device in nanometer scale in very challenging.
- Any abnormality of geometrical shape can result in defect.
- Defects are caused by:
  - process variation (channel length, transistor treshold voltage, metal interconnect width and thickness, intermetal layer dielectric thickness, …)
  - Random localized impefections (resistive bridging between metal lines, resistive opens in metal lines …)

Introduction

- Recent advances in physics, chemistry and material science have allowed production of nanometer-scale structures using sophisticated fabrication techniques.
- It is known that these devices have much higher manufacturing defect rates compared to conventional CMOS devices:
  - more sensitive to noise-induced errors, more susceptible to process variations, electromigration, material aging.
  - as the integration scale increases, more transistors can be fabricated on a single chip -> reducing the cost per transistor, but increases the difficulty of detecting faults produced by these defects.

Test Generation

- A **fault** is a representation of a defect reflecting a physical condition that causes a circuit to fail to perform a required operation.
- A **failure** is a deviation in the performance of a circuit from its specified behaviour.
- A circuit **error** is a wrong output signal produced by a defective circuit.
- To test a circuit with *m* inputs and *n* outputs, a set of input patterns (**test vectors**) is applied to the circuit under the test and its responses are compared to the known good responses of a fault-free circuit.

Test Generation

- It is difficult to know how many test vectors are required to guarantee a satisfactory reject rate.
- For *n* input circuit, we can apply 2^n possible input patterns – this is called **exhaustive testing**. This is not practical if *n* is large.
- And this also does not guarantee that all states are visited in the case of sequential circuits.
- Exhaustive testing forms the basis of **functional testing**, where every entry in the truth table for the combinational logic block is tested.

Test Generation

- A more practical approach is to select specific test patterns based on circuit structural information and a set of **fault models**.
- This approach is called **structural testing**.
- Structural testing saves time and improves test efficiency as the total number of test patterns is decreased because the test vectors target specific faults that would result from real defects in the manufactured circuit.
- Structural testing cannot guarantee detection of all possible manufacturing defects.

Test Generation

- The use of fault models does provide a quantitive measure of the fault-detection capabilities of a given set of test vectors for a targeted fault model.
- This measure is called a fault coverage defined by:

$$Fault\ coverage = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults}$$

- Because some fault may be undetectable, it is impossible to obtain FC = 100 %.
- **Undetectable fault**: there is no test that distinguish fault-free circuit and a circuit with this fault.

Test Generation

- Definition for FC can be modified to **fault detection efficiency** or **effective fault coverage**, defined by:

$$Fault\ detection\ effeciency = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults - number\ of\ undetectable\ faults}$$

- FC is linked to the yield and the defect level by:

$$Defect\ level = 1 - yield^{(1-\text{fault coverage})}$$

- Any input pattern (or a sequence of input patterns) that produces a different output response in a faulty circuit from that of the fault-free circuit is a **test vector.**

Test Generation

- The goal of test generation is to find an efficient set of test vectors that detects all faults considered for CUT.
- Because a given set of test vectors is usually capable of detecting many faults in a circuit, **fault simulation** is typically used to evaluate the FC obtained by that set of test vectors.
- Fault simulation is based on the same fault models as test generation.

Fault models

- Because of the diversity of VLSI defects, it is very difficult to generate tests for real defects.
- Fault models are necessary for generating and evaluating a set of test vectors.
- Good fault model should satisfy two criteria:
  - It should accurately reflect the behavior of defects.
  - It should be computationally efficient in terms of fault simulation and test pattern generation.
- Many fault models have been proposed so far, but no single fault model reflects the behaviour of all possible defects that can occur.

Fault models

- For a given fault model, there will be *k* different types of faults that can occur at *n* possible fault sites.
- Assuming that there is only one fault in the circuit, the total number of possible single faults, referred as **single-fault model** or **single-fault assumption**, is

$$Number\ of\ single\ faults = k \times n$$

- In reality, multiple faults may occur in the circuit. The total number of multiple faults, referred to as the multiple-fault model, is given by:

$$Number\ of\ multiple\ faults = (k+1)^n - 1$$

Fault models

- While the number of multiple-fault model is more accurate than the single-fault assumption, the number of possible faults becomes impractically large.
- It has been shown that high FC obtained under the single-fault assumption results in high FC for the multiple-fault model -> single-fault assumption is typically used for test generation.
- Two or more faults may result in identical faulty behaviour for all possible input patterns. These faults are called **equivalent faults** and can be represented by any single fault from the set of equivalent faults.

Fault models

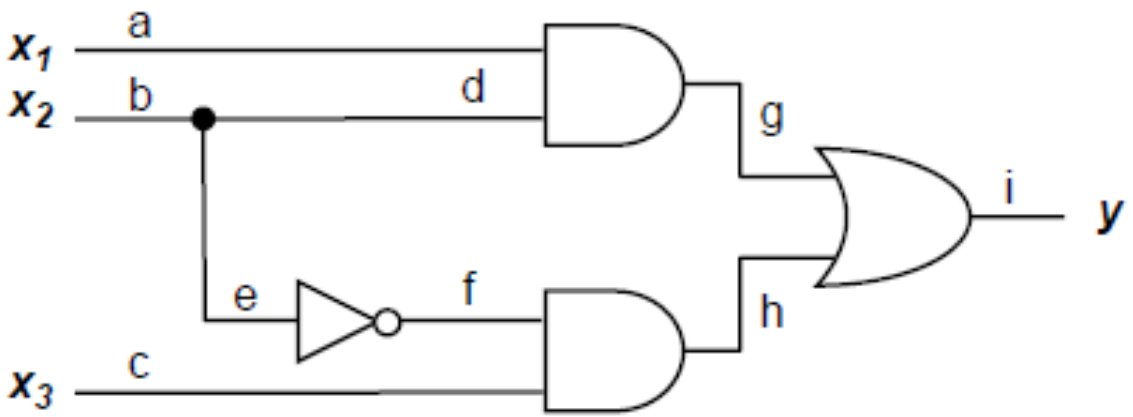
- As a result, the number of single faults to be considered for test generation for the given circuit is much less than k x n.
- The reduction of entire set of single faults by removing equivalent faults is referred to a **fault collapsing**.
- Fault collapsing helps to reduce both test generation and fault simulation times.

Stuck-at Faults

- One of the most useful models for fault modeling.
- A stuck-at fault transforms the correct value on the faulty signal line to be stuck at a constant logic value, either logic 0 or a logic 1, referred to as **stuck-at-0** or **stuck-at-1**.
- Stuck-at fault affects the state of logic signals on all lines in a logic circuit:
  - primary inputs (PI)
  - primary outputs (PO),
  - internal gate inputs and outputs,
  - fanout stems,
  - fanout branches.

Stuck-at Faults

- Consider the example circuit, where 9 signal lines represent potential fault sites.



- There are 2 x 9 = 18 possible faulty circuits under the single fault assumption.
- The next table gives the truth table for the fault-free and faulty circuits for all possible single stuck-at faults.

Stuck-at Faults

Truth table for the fault-free and faulty circuit.

$x_1x_2x_3$	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**

Stuck-at Faults

- All faults but d SA1, e SA0 and f SA1 are detected with two or more test vectors -> test vectors 011 and 100 must be included in any set of test vectors that obtain 100 % FC.
- These two test vectors detect ten faults and the remaining eight faults can be detected with test vectors 001 and 110.
- These four test vectors obtain 100 % FC for this circuit.
- Four sets of equivalent faults can be observed -> one fault from each set can be used to represent all of the equivalent faults in that set.

Stuck-at Faults

- There is a total of ten unique faulty responses to the complete set of input test patterns -> these ten faults constitute the set of collapsed faults for the circuit.
- Stuck-at fault collapsing typically reduces the total number of faults by 50 to 60 %.
- Fault collapsingis based on the fact that a SA0 at the input of AND (NAND) gate is equivalent to SA0 (SA1) at the output of the gate.
- Similar for SA1 for OR (NOR) gates and SA0 (SA1) for the inverter.
- Also, SA fault at the output of the gate is in fanout-free circuit equivalent to the same SA fault at the input of the driven gate.

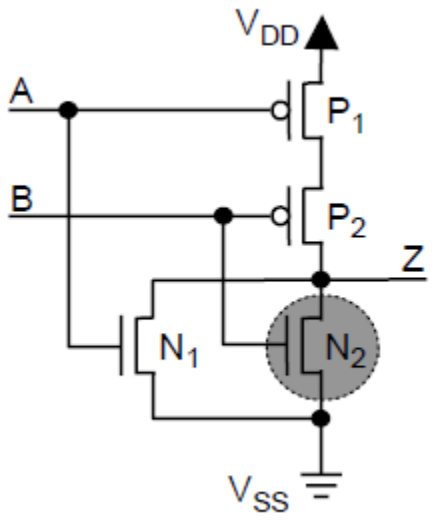
Number of collapsed stuck-at faults

$$\begin{aligned} \textit{Number of collapsed faults} = {} & 2 \times (\textit{number of POs} + \textit{number of fanout stems}) \\ & + \textit{total number of gate (including inverter) inputs} \\ & - \textit{total number of inverters} \end{aligned}$$

- In the example shown: 2 x (1 + 1) + 7 – 1 = 10
- All single-input gates are treated the same as inverter in calculation.
- Theorem 1:
  - A set of test vectors that detects all single stuck-at faults on all Pis of a fanout-free combinational logic will detect all single stuck-at faults in that circuit.
- Theorem 2:
  - A set of test vectors that detect all single stuck-at faults on all Pis and all fanout branches of a combinational circuit will detect all single stuck-at faults in that circuit.

Transistor Faults

- Stuck-at fault model cannot accurately model transistor at the switch level -> transistor can be **stuck-open** or **stuck-short**, referred also a **stuck-off** or **stuck-on**.
- Example, consider N_2 is stuck-open:



Transistor Faults (Stuck-Open)

- If AB = 01 is applied, output Z should be 0, but the stuck-open fault casuses Z to be isolated from ground.
- Output Z keeps previous state (P_2 and N_1 are off).
- Sequence of test vectors AB = 00 -> 01 is required to detect this fault.
- Fault-free circuits Z = 1 -> 0
- Faulty circuit: Z = 1 -> 1
- Stuck-open fault in a CMOS combinational circuit requires a sequence of two vectors for detection as opposed to a single-test vector for stuck-at fault model.

Transistor Faults (Stuck-Short)

- Stuck-short results in conducting path between V_{dd} and V_{ss}.
- If N2 is stuck-short, there is a conducting path for AB = 00.
- This creates a voltage divider at the output node Z -> logic value is a function of the resistors of both condusting transistors and cannot be necessary interpreted as an incorrect logic value.
- Stuck-short faults can be detected by monitoring the power supply current during steady state, referred as I_{DDQ}.
- This technique for monitoring the the power supply current is referred to as **I_{DDQ} testing**.

Transistor Faults

- This example has a total of 2 x 4 =8 possible transistor faults.
- There are also equivalent faults at the transistor level, for example stuck-open faults for P1 and P2 are indistinguisahble.
- The same for stuck-short faults for N1 and N2.
- The number of collapsed transistor faults is:

$$Number\ of\ collapsed\ faults = 2 \times T - T_S + G_S - T_p + G_p$$

  - Example: 2 x 4 – 2 + 1 – 2 + 1 = 6.

Transistor Faults

- Behaviour of fault-free circuit and each of 8 possible faulty circuits.
- Test vectors are the same for stuck-open faults for P1 and P2 and stuck-short faults for N1 and N2.

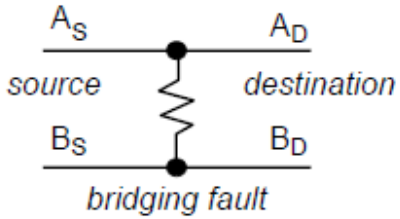
AB	00	01	10	**11**
Z	1	0	0	**0**
N_1 stuck-open	1	0	Last Z	**0**
N_1 stuck-short	I_{DDQ}	0	0	**0**
N_2 stuck-open	1	Last Z	0	**0**
N_2 stuck-short	I_{DDQ}	0	0	**0**
P_1 stuck-open	Last Z	0	0	**0**
P_1 stuck-short	1	0	I_{DDQ}	**0**
P_2 stuck-open	Last Z	0	0	**0**
P_2 stuck-short	**1**	**I_{DDQ}**	**0**	**0**

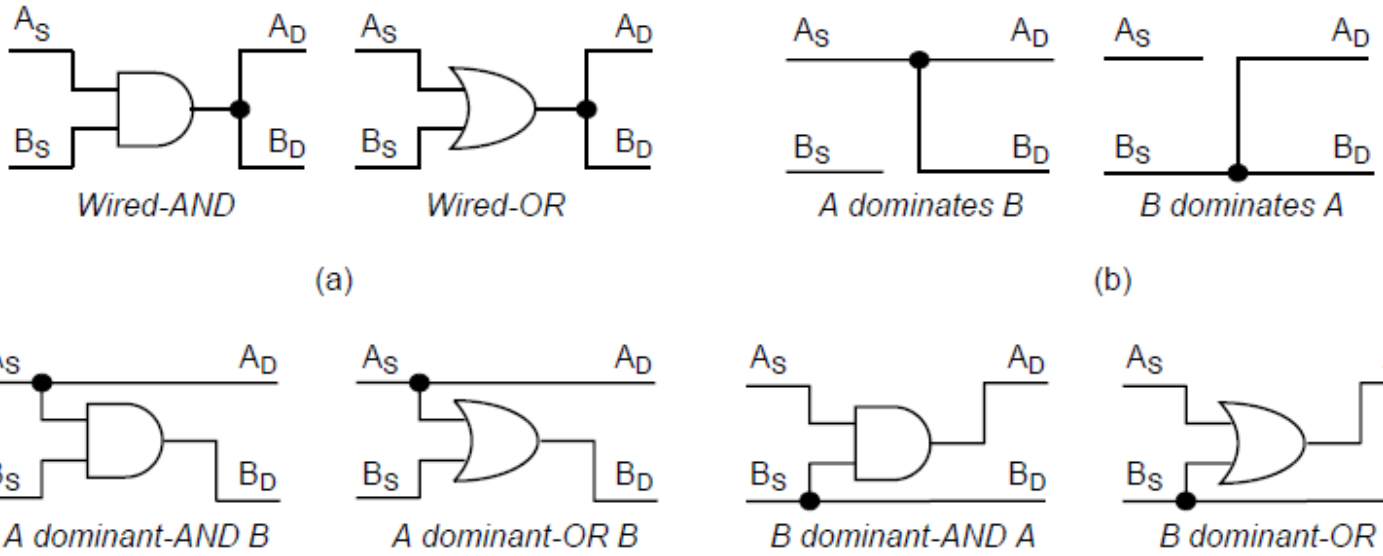
Opens and Short Faults

- Defects in VLSI devices include opens and shorts in the wires that interconnect the transistors and logic gates.
- Opens in wires behave similar to transistor stuck-open faults.
- Short between two elements (transistor terminals or connections between transistors and gates) is referred to as a **bridging fault**.
- Short to Vdd or Vss is stuck-at-1 or stuck-at-0.
- If two connections are bridged, the most common model is **wired-AND** or **wired-OR** bridging fault model.

Opens and Short Faults

- The wired-AND bridging fault means the signal net formed by the two shorted lines will take on a logic 0 if either shorted line is sourcing a logic 0. Similar for 1 for wired-OR bridging fault.





Opens and Short Faults

- The truth tables for fault-free and faulty behaviour of bridging faults is:

$A_S B_S$	0	0	0	1	1	0	1	1
$A_D B_D$	0	0	0	1	1	0	1	1
Wired-AND	0	0	0	0	0	0	1	1
Wired-OR	0	0	1	1	1	1	1	1
A dominates B	0	0	0	0	1	1	1	1
B dominates A	0	0	1	1	0	0	1	1
A dominant-AND B	0	0	0	0	1	0	1	1
B dominant-AND A	0	0	0	1	0	0	1	1
A dominant-OR B	0	0	0	1	1	1	1	1
B dominant-OR A	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**

Opens and Short Faults

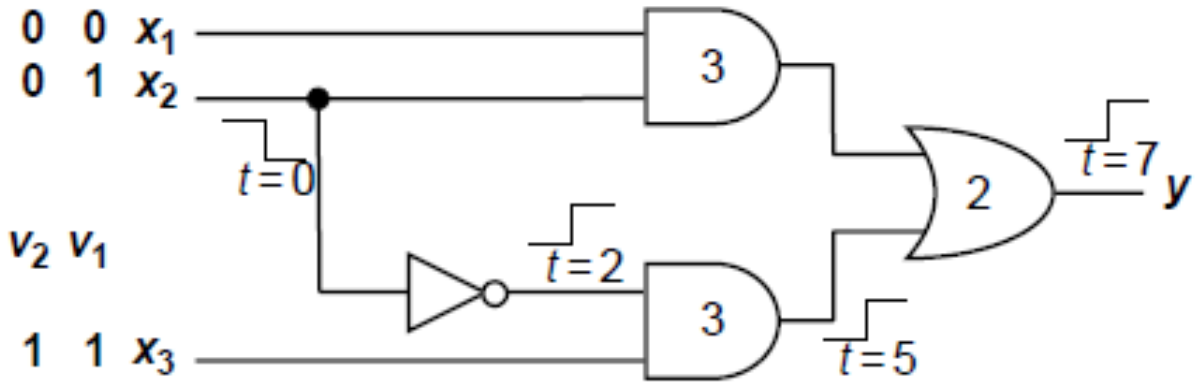
- Bridging faults commonly occur in practice and they can be detected by I_{DDQ} testing.
- Many bridging faults are detected by a set of test vectors that obtains high stuck-at FC.
- In the presence of a bridging fault, a combinational circuit can have a feedback path and behave like a sequential logic circuit -> more complicated test.
- Number of possible bridging faults for a circuit with N signal nets is N x (N – 1)/2, but a bridging fault between two nets on opposite sites of device is not possible -> bridging faults are considered on the extracted signal nets (after physical design)

Delay Faults and Crosstalk

- A **delay fault** causes excessive delay along a path such that the total propagation delay falls outside the specified time limit.
- There are different delay fault models:
  - Gate-delay fault,
  - Transition fault,
  - Path-delay fault
- Path-delay considers the cumulative propagation delay along a signal path through CUT – sum of all gate delays along the path.

Delay Faults and Crosstalk

- As with transistor stuck-open faults, delay faults require an ordered pair of test vectors to sensitize a path through the logic circuit and to create a transition along that path in order to measure a path delay.
- Example:



Delay Faults and Crosstalk

- The use of nanometer techniques increases cross-coupling capacitances and inductance between interconnections, leading to sever crosstalk effects that may result in improper functioning of the chip.
- Crosstalk effects can be divided into two categories:
  - crosstalk glitches,
  - crosstalk delays.

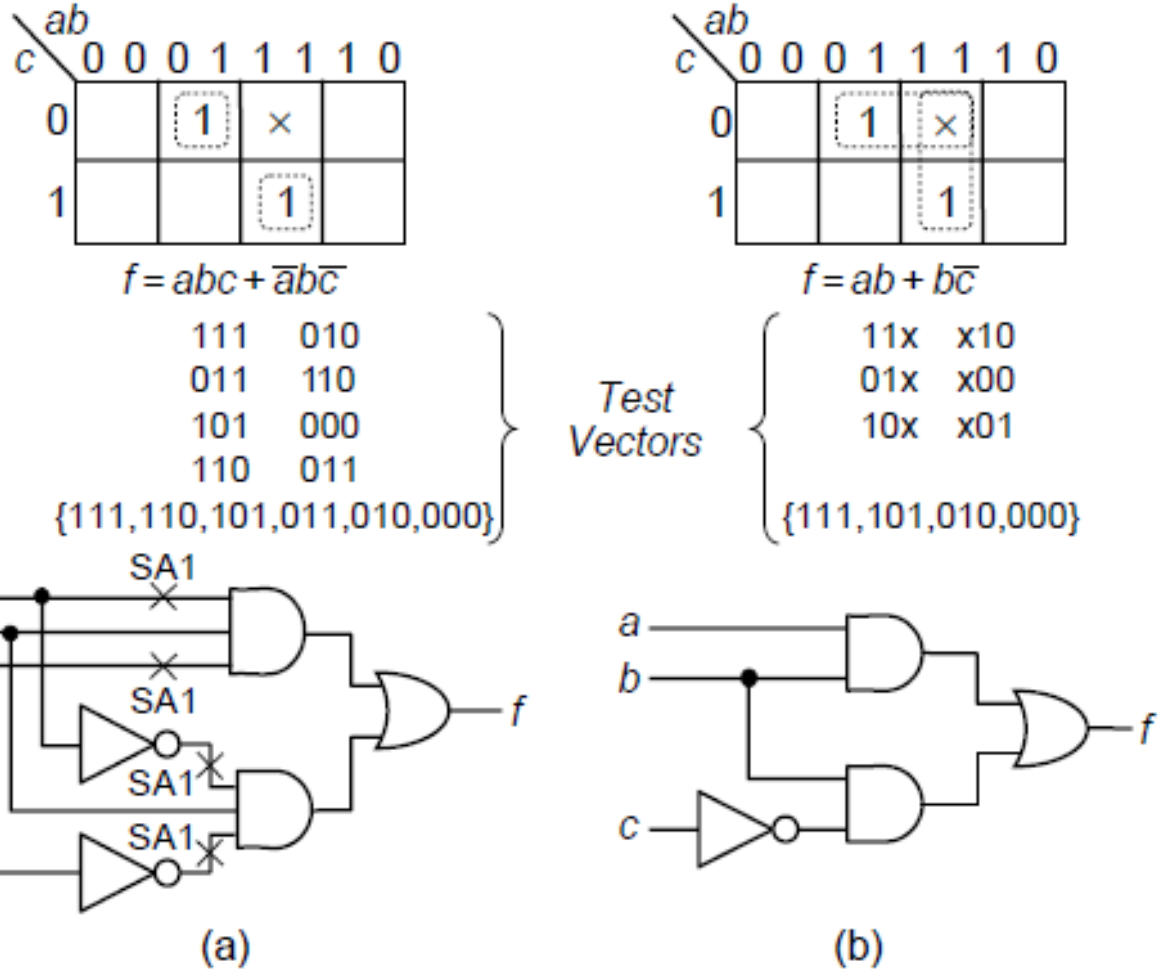
Analog Fault Models

- Typical analog fault models include shorts, opens and parameter variations in both active in passive components.
- Short and opens usually results in **catastrophic faults**.
- Parameter variations that cause components to be out of their tolerance ranges result in **parametric faults**.
- It is very difficult to identify critical parameters and a model of process fluctuations.
- Because of the complex nature of analog circuits, direct applications of digital fault models (except shorts and opens) is inadequate in capturing faulty behaviour in analog circuits.

RTL and Behavioral Level

- FC obtained at RTL is lower than at the gate level since the implementation details at RTL are either unknown or subject to change.
- Example: $f = \bar{a}b\bar{c} + abc + xab\bar{c}$
- Don‘t care x can be treated as 0 (a) or 1 (b).
- Set of test vectors in (b) is a subset of test vectors in (a) and does not detect the four SA1 faults shown in (a).

RTL and Behavioral Level

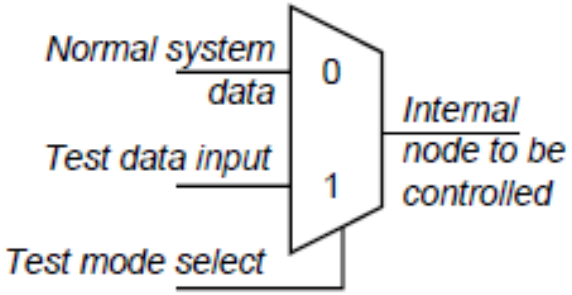


Design for Testability

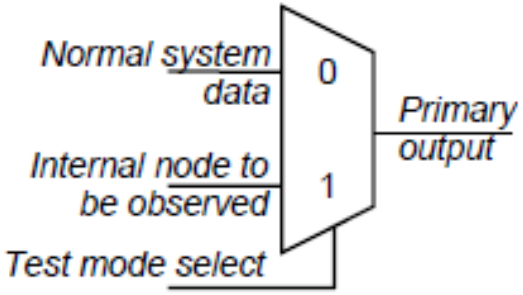
- In order to test a circuit, we have to control and observe logic values of internal lines.
- **Design for testability** (**DFT**) techniques are used to overcome this obstacle.
- DFT can be devided to:
  - Ad hoc DFT techniques
  - Level-sensitive scan design (LSSD), scan design
  - Built-in self test (BIST)

Ad hoc methods

- The goal is to target only those portions of the circuit that would be difficult to test and to add circuitry to improve controllability and observability.
- Ad hoc techniques use test point insertion to access internal nodes directly.
- Most common use of test point is the multiplexer.



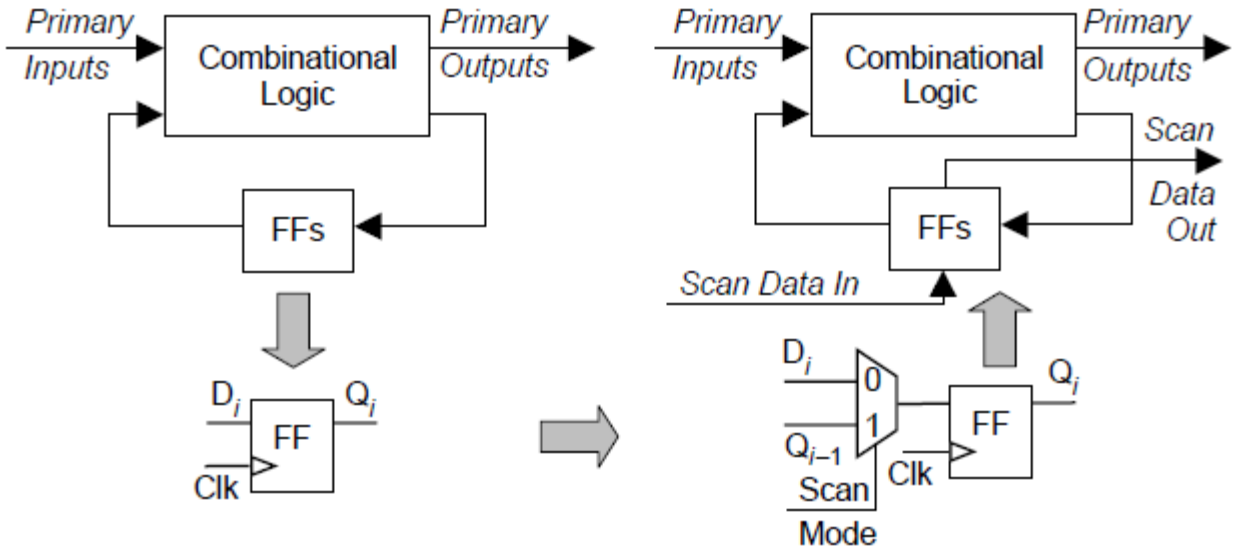
(a) controllability test point



(b) observability test point

Level-sensitive can design (scan design)

- Testability is improved by adding extra logic to each FF in the circuit to form a shift register or scan chain.
- The problem of testing sequential logic is reduced to that of testing combinational logic and ATPG systems developed for combinational logic.



Built-in-self-test (BIST)

- BIST integrates a test pattern generator (TPG) and an ouput response analyzer (ORA) in the VLSI device to perform testing internal to the IC.
- Test circuitry is within the CUT -> BIST can be used at all levels of testing (wafer, system-level).