Testing and Design-for-Testability (DFT) for Digital Integrated Circuits

Hafizur Rahaman

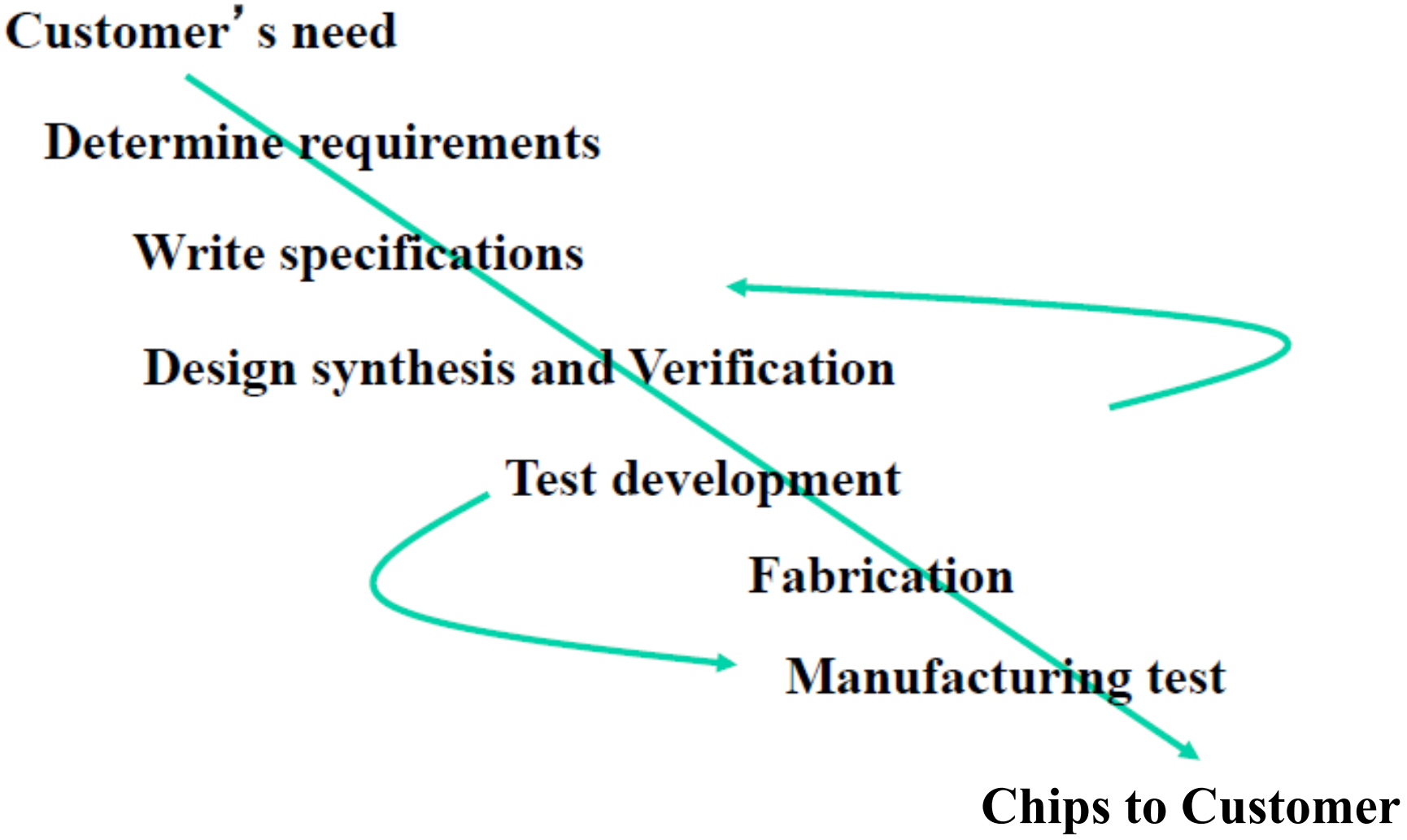
(hafizur@vlsi.iiests.ac.in)

School of VLSI Technology
Indian Institute of Engineering Science and Technology (IIEST), Shibpur
India

Outline of the Talk

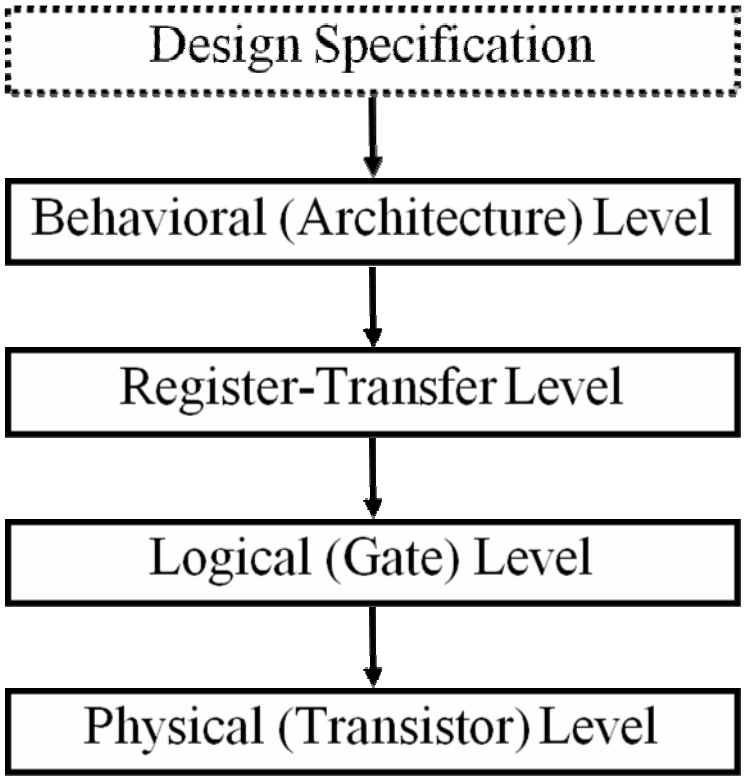
- Introduction
- Test methods
- Design for testability
- Some emerging technologies

VLSI Realization Process



Design synthesis

- Design synthesis: Given an I/O function, develop a procedure to manufacture a device using known materials and processes.
- Different levels of abstraction during design
- CAD tools used to synthesize design from RTL to physical level



Verification and Test

Verification:

- Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given I/O function.

Test

- A manufacturing step that ensures that the physical device, manufactured from the synthesized design, has no manufacturing defect.
- to test the behavioral correctness of a VLSI design
- process by which a defect in the circuit or system can be exposed.

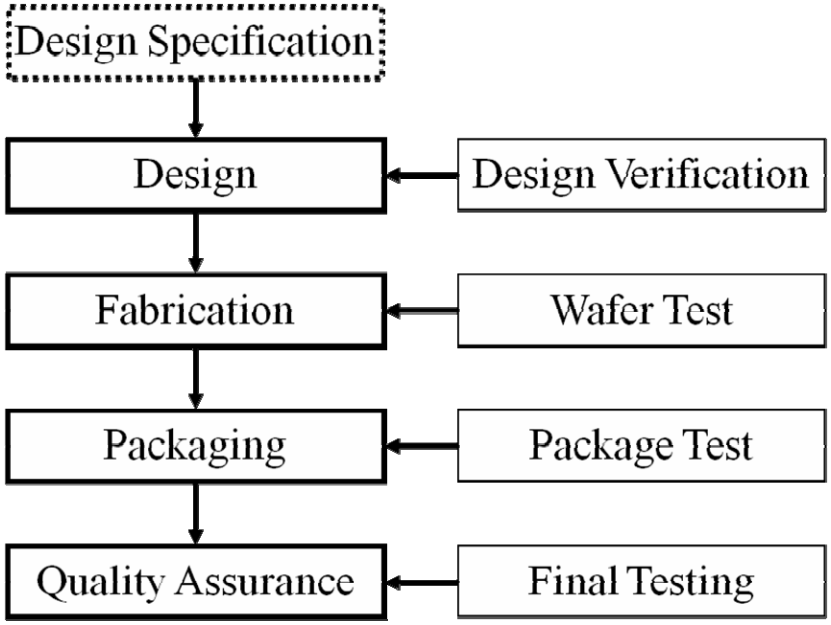
Verifications vs. Test

- Verifies correctness of design.
- Performed by simulation, hardware emulation, or formal methods.
- Performed once prior to manufacturing.
- Responsible for quality of design.

- Verifies correctness of manufactured hardware.
- Two-part process:
  - 1. Test generation: software process executed once during design
  - 2. Test application: electrical tests applied to hardware
- Test application performed on every manufactured device.
- Responsible for quality of devices.

Testing in VLSI Design Cycle

- Design verification targets design errors
  - Corrections made prior to fabrication
- Remaining tests target manufacturing defects



Detection of Defects

- Detection of defects may be done in three phases:
- Design verification involves ascertaining logical correctness and timing behavior of the circuit through simulation
- Manufacturing tests check for the specific types of defects produced during fabrication.
- Field test (day-to-day testing) – detects the systems when the system is in the field.

Definitions (Defect, Fault and Error)

- Defect: refers to a physical imperfection in the circuit or system
- Fault : an actual defect that occurs in digital circuit or device.
- When a vector is applied to the faulty circuit which produces an incorrect response, an error is said to have occurred.

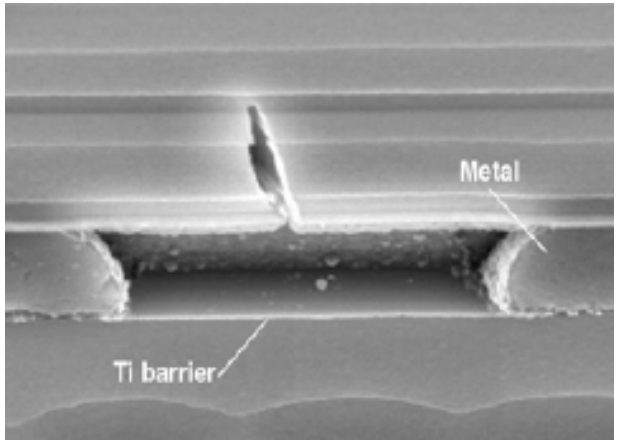
Fault: logical fault and parametric.

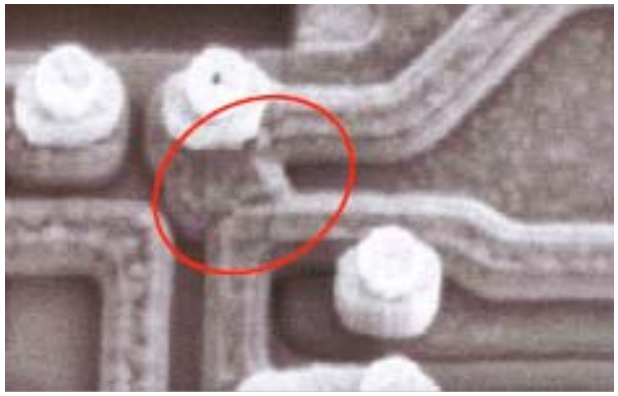
- A fault which can change the logic value on a line in the circuit from logic 0 to logic 1 or vice versa is called a logical fault.

- if the fault causes some parameters of the circuit to change, such as the current drawn by the circuit, then it is termed parametric.

Some Real Defects in Chips

- **Processing defects**
  - Missing contact windows
  - Parasitic transistors
  - Oxide breakdown
  - . . .
- **Material defects**
  - Bulk defects (cracks, crystal imperfections)
  - Surface impurities (ion migration)
  - . . .
- **Time-dependent failures**
  - Dielectric breakdown
  - Electro-migration
  - . . .
- **Packaging failures**
  - Contact degradation
  - Seal leaks
  - . . .





Classification of Faults

- **Transient fault:** A fault is called transient if it is only present for a small duration.
- **Intermittent fault:** A fault is intermittent if it appears regularly but is not present continuously.
- **Permanent fault**: If a fault is present continuously, it is called permanent.

Transient Faults:

- have been the dominant cause of system failures.
- may be caused by α-particle radiation, power supply fluctuation, etc.
- No permanent damage is done by these faults.
- are hard to detect because of their short duration.

Classification of Faults

- **Intermittent Faults:**
- are also difficult to detect and locate.
- can be caused by
  - loose connections,
  - bad designs,
  - environmental effects like temperature and humidity variations.
- **Permanent Faults:**
- are the easiest to detect.
- are predominantly caused by
  - shorts and opens in VLSI circuits

Fault Detection and Fault Location

- **Fault detection:** the discovery of something wrong in a digital system or circuit.

- **Fault location:** the identification of the faults with components, functional modules, or subsystems, depending on the requirements. It is very difficult to locate a fault.

- **Fault diagnosis**: includes both fault detection and fault location.

Why Testing is important?

- Moore's law results in the steady decrease of dimensions (feature size) of the transistors and interconnecting wires
- reduction in feature size increases the probability of a manufacturing defect in the IC
- A very small defect can easily result in a faulty transistor or interconnecting wire when the feature size is less than 100 nm.
- Circuits are used in highly sophisticated applications
- A single failure may cause large deviation from the expected performance

Problems of Ideal Tests

- Ideal tests detect all defects produced in the manufacturing process.
- Ideal tests pass all functionally good devices.
- Very large numbers and varieties of possible defects need to be tested.
- Difficult to generate tests for some real defects.

Defect-oriented testing is an open problem.

Real Tests

- Based on analyzable fault models, which may not map on real defects.
- Incomplete coverage of modeled faults due to high complexity.
- Some good chips are rejected. Fraction (or percentage) of such chips is called the *yield loss*.
- Some bad chips pass tests. Fraction (or percentage) of bad chips among all passing chips is called the *defect level*.

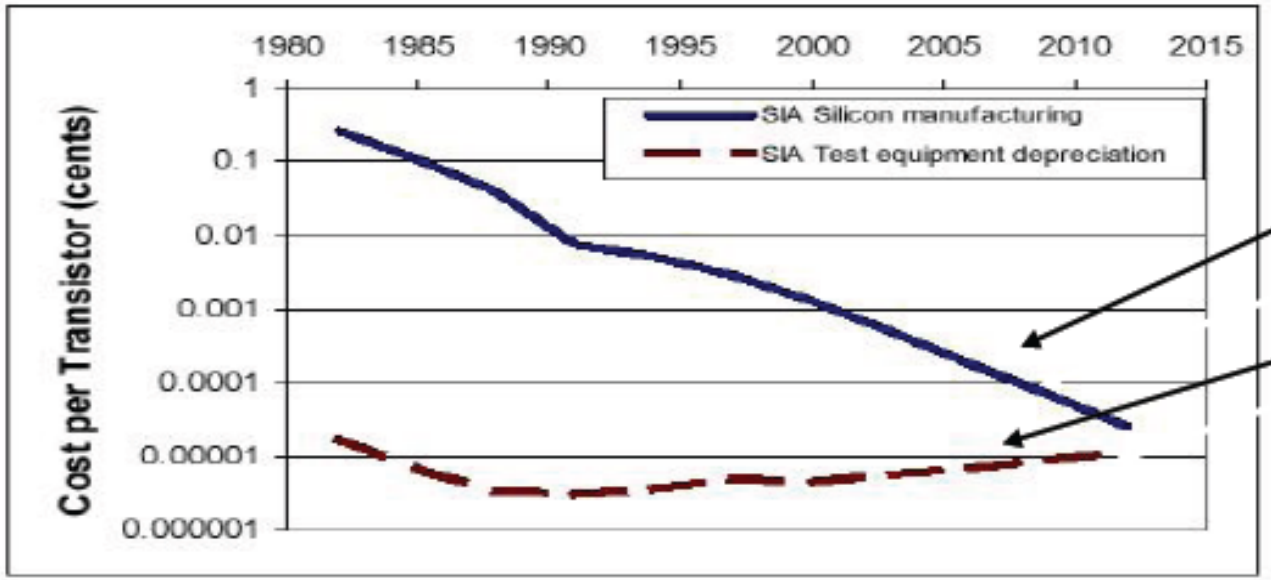
Yield and Rejection Rate

- Yield = (Number of acceptable chips)/(Total number of chips fabricated)
- A good device fails the test and appears as faulty. Fraction (or percentage) of such chips is called the yield loss.
  - Catastrophic yield loss: is due to random defects.
  - Parametric yield loss: is due to process variations.
- A faulty device appears to be a good chips passing the test.

Reject rate or defect level = (Number of faulty chips passing final test)/Total number of chips passing final test

Cost of Test

International Technology Roadmap for Semiconductors
ITRS 2.0 available at http://www.itrs2.net



Manufacturing Cost

Test Cost

Cost of Test

- Emergence of more advanced ICs and SOC semiconductor devices is causing test costs to escalate to as much as 50 percent of the total manufacturing cost.

  – M. Kondrat, Electronic News, Sept 9, 2002.

- As a result, semiconductor test cost continues to increase in spite of the introduction of DFT, and can account for up to 25-50% of total manufacturing cost.

  – T. Cooper, G. Flynn, G. Ganesan, R. Nolan, C. Tran, Motorola

- Test may account for more than 70% of the total manufacturing cost – test cost does not directly scale with transistor count, dies size, device pin count, or process technology

  - ITRS 2003.

Cost of Manufacturing Testing

- 0.5-1.0GHz, analog instruments,1,024 digital pins:

  ATE purchase price

  – = $1.2M + 1,024 x $3,000 = $4.272M

- Running cost (five-year linear depreciation)

  – = Depreciation + Maintenance + Operation

  – = $0.854M + $0.085M + $0.5M

  – = $1.439M/year

- Test cost (24 hour ATE operation)

  – = $1.439M/(365 x 24 x 3,600)

  – = 4.5 cents/second

Top Chip Manufacturers (2013)

Source : iHS iSuppli Semiconductor rankings for 2013 (foundries excluded)

Rank 2013	Rank 2012	Company	Revenue (million $USD)	2013/2012 changes	Market share
1	1	Intel(1)	46 960	-1.0%	14.8%
2	2	Samsung(2)	33 456	+7.0%	10.5%
3	3	Qualcomm	17 341	+31.6%	5.5%
4	10	Micron Technology(3)	14 168	+109.2%	4.5%
5	7	SK Hynix	13 335	+48.7%	4.2%
6	5	Toshiba	12 459	+11.9%	3.9%
7	4	Texas Instruments	11 379	-5.5%	3.6%
8	9	Broadcom	8 121	+3.5%	2.6%
9	8	STMicroelectronics	8 076	-4.9%	2.5%
10	6	Renesas Electronics(4)	7 822	-15.3%	2.5%

(1) Intel Corporation's acquisition of Fujitsu Semiconductor Wireless Products.
(2) Samsung Electronics sold their 4- and 8-bit microcontroller business to IXYS.
(3) Micron Technology's acquisition of Elpida.
(4) Broadcom's acquisition of Renesas Electronics' LTE unit.

Top Chip Manufacturers (2014)

Source : iHS iSuppli Semiconductor rankings for 2014 (foundries excluded)

Rank 2014	Rank 2013	Company	Revenue (million $USD)	2014/2013 changes	Market share
1	1	Intel	49 964	+6.3%	14.1%
2	2	Samsung Electronics	38 273	+15.6%	10.8%
3	3	Qualcomm	19 266	+11.9%	5.5%
4	4	Micron Technology	16 389	+16.1%	4.6%
5	5	SK Hynix	15 737	+22.9%	4.5%
6	6	Texas Instruments	11 420	+6.8%	3.5%
7	7	Toshiba	8 496	-9.6%	2.4%
8	8	Broadcom	8 387	+2.5%	2.4%
9	9	STMicroelectronics	7 395	-8.5%	2.1%
10	15	MediaTek	7 194	+57.5%	2.0%

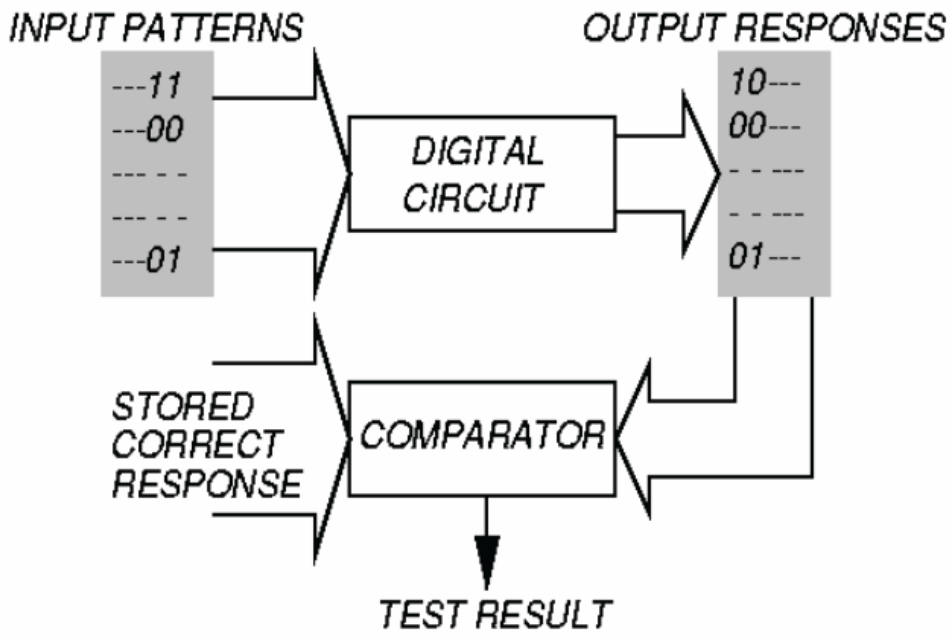
Roles of Testing

- Detection: Determination whether or not the *device under test* (DUT) has some fault.
- Diagnosis: Identification of a specific fault that is present on DUT.
- Device characterization: Determination and correction of errors in design and/or test procedure.
- *Failure mode analysis* (FMA): Determination of manufacturing process errors that may have caused defects on the DUT.

VLSI Testing Process

- Two processes: test generation and test application.
- Goal of test generation is to produce test patterns for efficient testing.
- Test application is the process of applying those test patterns to the CUT and analyzing the output responses
  - If incorrect (fail), CUT assumed to be faulty
  - If correct (pass), CUT assumed to be fault-free

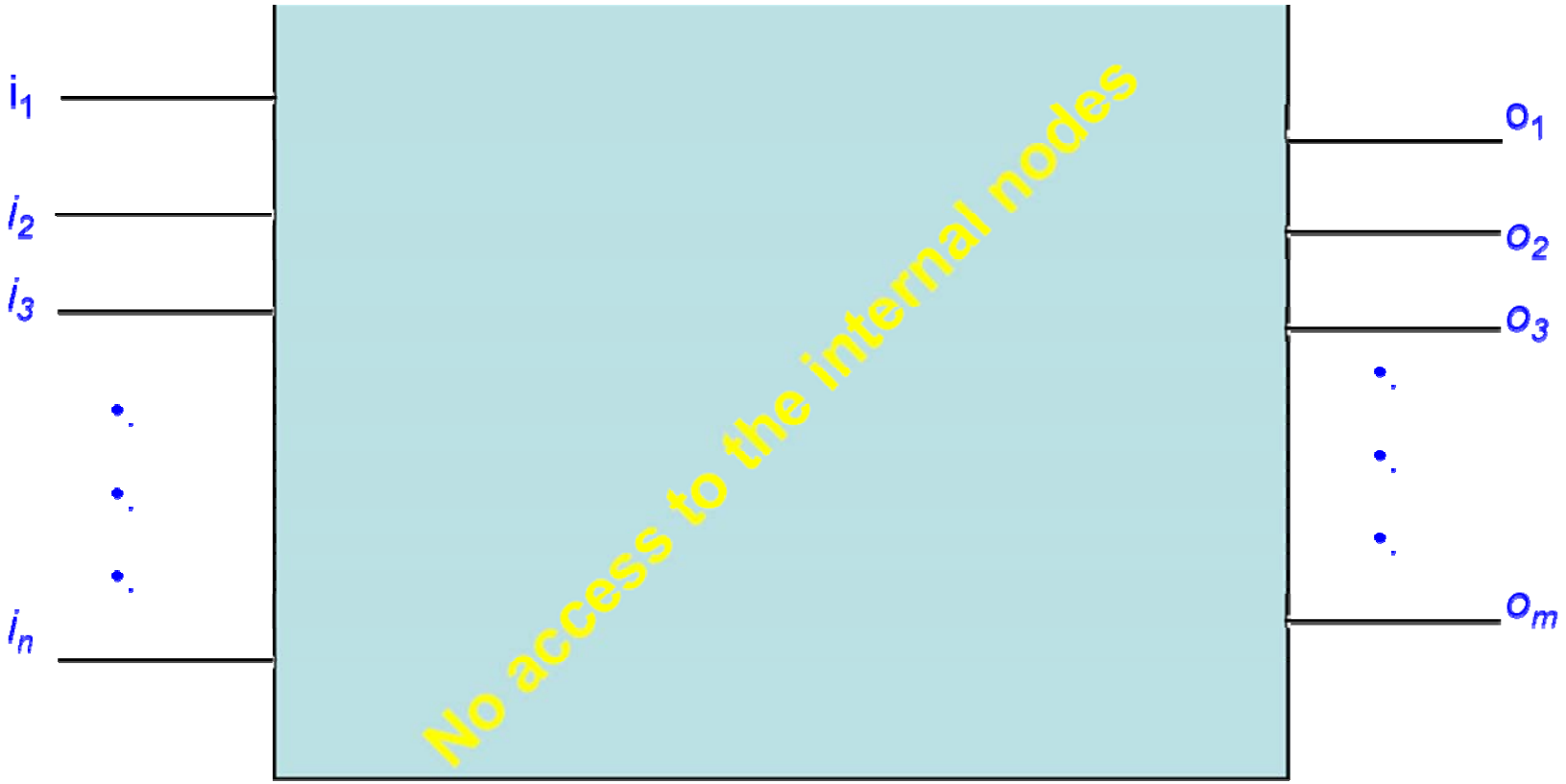
- Test application is performed by either *automatic test equipment (ATE)* or test facilities in the chip itself



Test Generation

- *Goal*: find efficient set of test vectors with maximum fault coverage
- No single fault model works for all possible defects
- A certain amount of test vectors based on fault models is applied for detection of the faults in VLSI Circuits.

Test Generation



- Apply all possible input patterns (2^n)
- For n=50, a test equipment operating at 1 MHz will take about 9500 months!

Fault Modeling

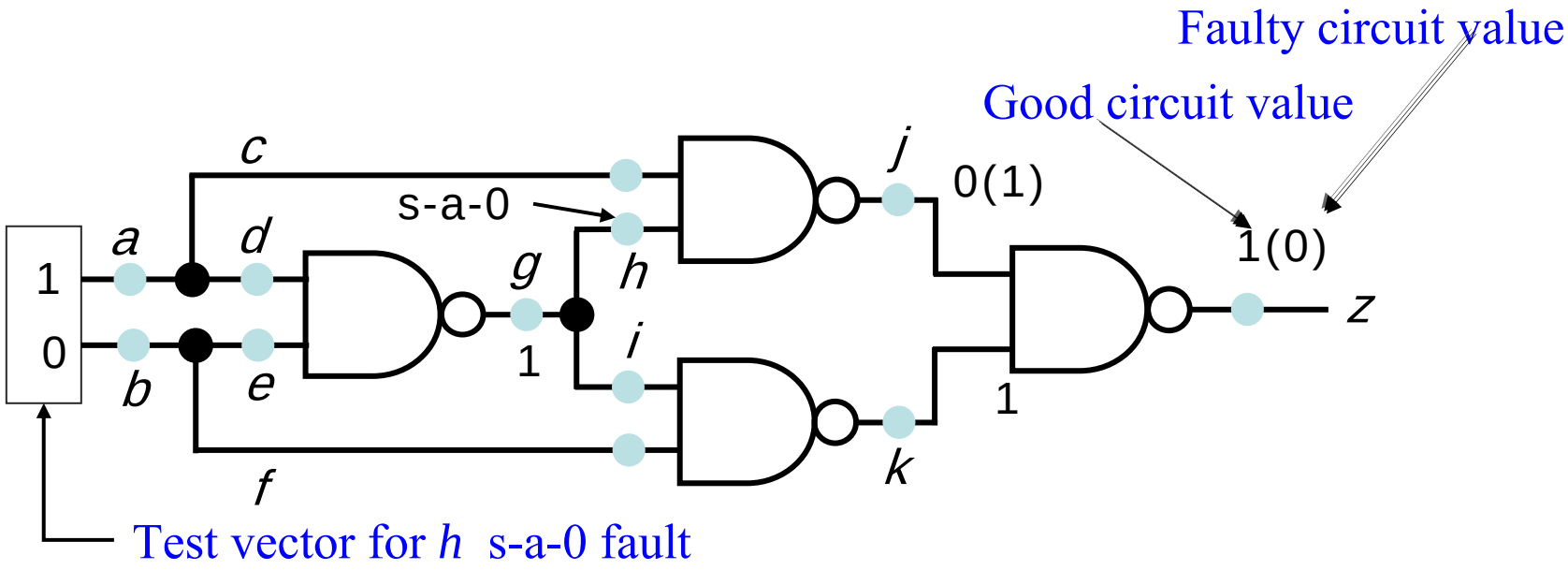
- Numerous possible physical failures in a large circuit
- Very difficult to detect a physical failure
- Many physical failures have the same effect on the logic
- We need to consider only the effect of physical failures on the logic
- Effects of physical failures are described at higher level : fault Model

Common Fault Models

- Stuck-at faults
- Transistor (Switch) faults
  - Stuck-open faults
  - Stuck-on or short faults
- Bridging Faults
- Delay faults (transition, path)

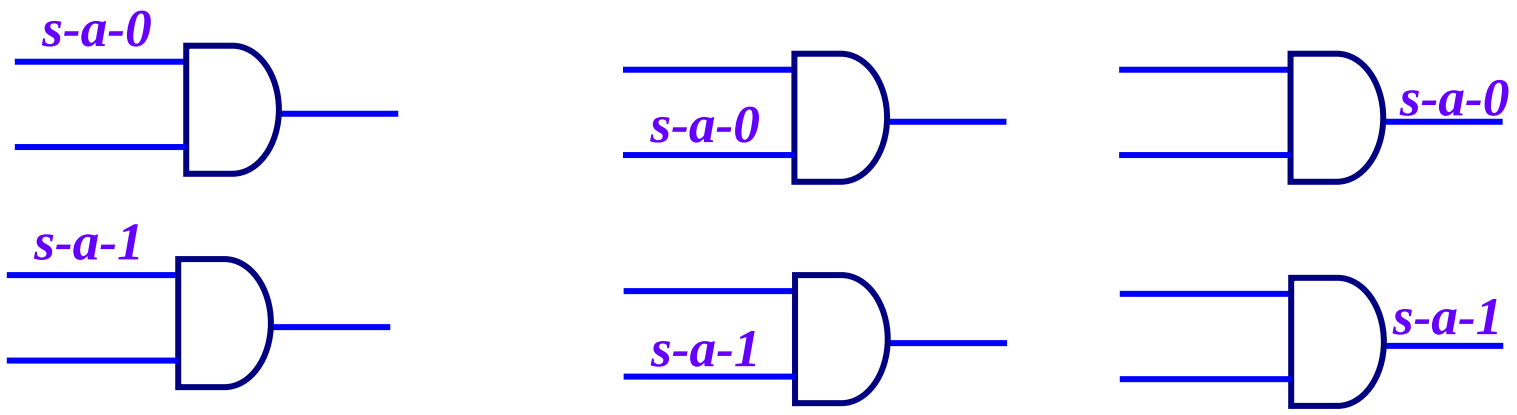
Single Stuck-at Fault

- Three properties define a single stuck-at fault
  - Only one line is faulty
  - Faulty line is permanently set to 0 or 1
  - Fault can be at an input or output of a gate
- Example: XOR circuit has 12 fault sites and 24 single stuck-at faults



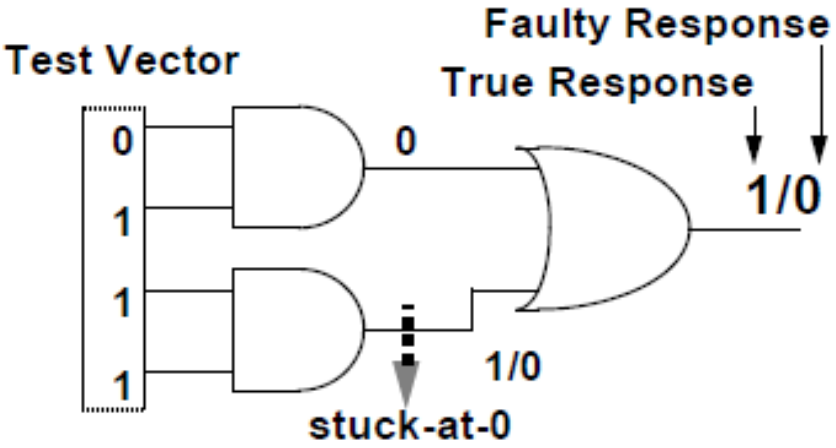
Single Stuck-at Fault

- Assumption
  - Only one line is Faulty
- 2n possibilities

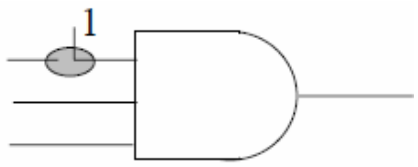


Single Stuck-at Fault

- **More Example**



Single Stuck-at (Stuck-Line (SSL)) Fault

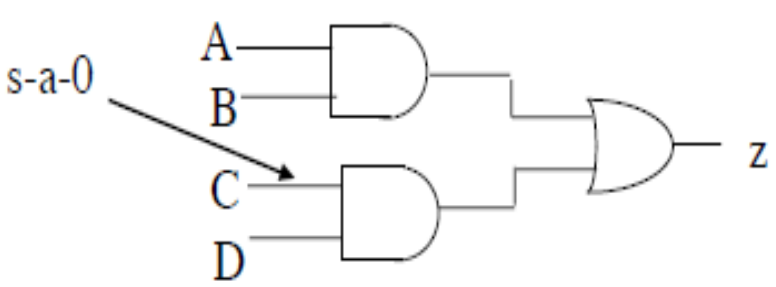


Any line in a logic circuit can be permanently stuck-at-1 (s-a-1, s/1) or stuck-at-0 (s-a-0, s/0)

- *Advantages*:
  - Also called *stuck-at*
  - Matches circuit level, easy to use
  - Moderate number of faults ($2n$ for an n-line circuit)
  - Tests for SSL faults provide good defect coverage (experiments)
- *Disadvantages*:
  - Does not account for timing/delay faults
  - Few physical defects behave like SSL faults

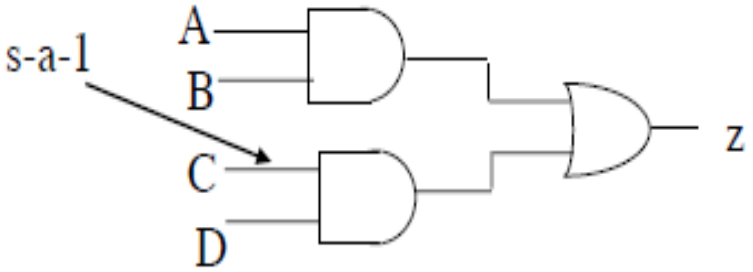
Single Stuck-at (Stuck-Line) Fault

- A single node in the circuit is stuck-at 1 (s-a-1) or 0 (s-a-0)



Fault-free function z = AB+CD

Faulty function z^f = AB

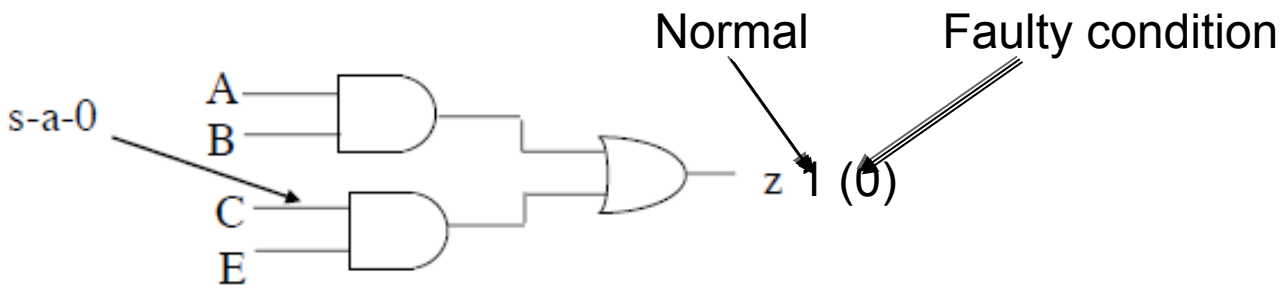


Fault-free function z = AB+CD

Faulty function z^f = AB+D

Single Stuck-at (Stuck-Line) Fault Detection

- A test pattern for s-a-*d* fault on *x* line is an input combination that: 1) places $\bar{d}$ on x (activation), 2) propagates fault effect to primary output



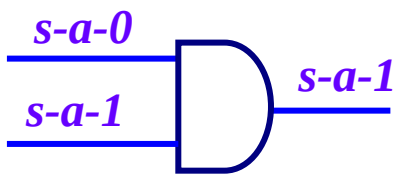
- ABCE = 0011 is a test pattern for s-a-0 on C

Multiple Stuck-at (Stuck-Line) Fault

- A multiple stuck-at fault means that any set of lines is stuck-at some combination of (0,1) values.
- The total number of single and multiple stuck-at faults in a circuit with k single fault sites is 3^k-1.
- A single fault test can fail to detect the target fault if another fault is also present, however, such masking of one fault by another is rare.
- Statistically, single fault tests cover a very large number of multiple faults.

Multiple Stuck-at (Stuck-Line) Fault

- **Assumption:** Several stuck-at faults can be simultaneously present
- 3^k - 1 possibilities



26 possibilities

Multiple Stuck-at (Stuck-Line) Fault

- In a fanout-free circuit, any complete test set for SSL faults detects all double and triple faults, and there exists a complete test set for SSL faults that detects all MSF faults

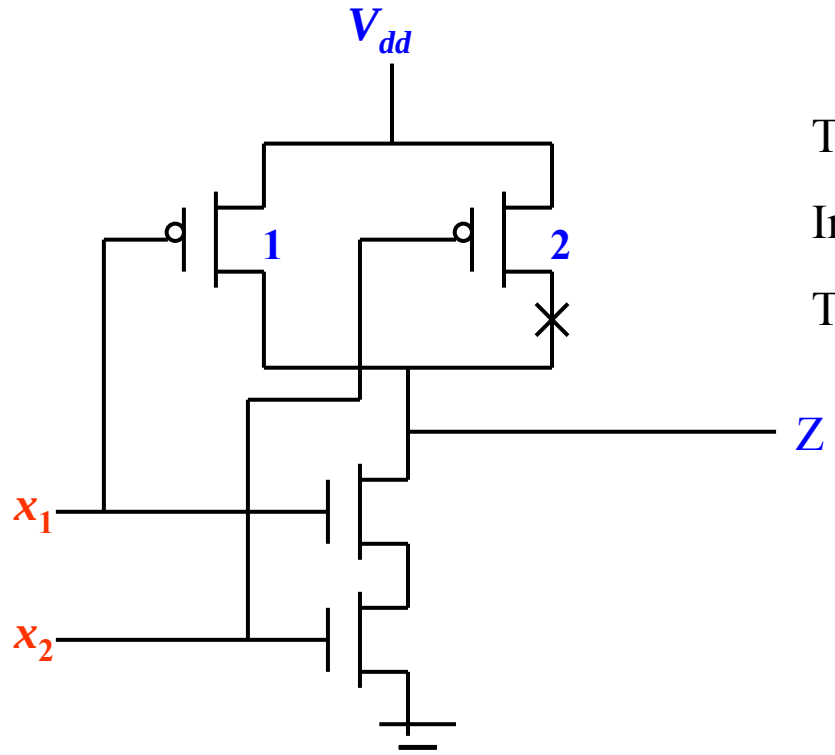
- In an internal fanout-free circuit, any complete test set for SSL faults detects at least 98% of all MSF faults of multiplicity less than 6.

Transistor (Switch)-Level Fault Models

- MOS transistor is considered an ideal switch and two types of faults are modeled:
  - Stuck-open - a single transistor is permanently stuck in the open state.
  - Stuck-short - a single transistor is permanently shorted irrespective of its gate voltage
- Detection of a stuck-open fault requires two vectors
- Detection of a stuck-short fault requires the measurement of quiescent current (I*DDQ*)

Stuck-open Fault

- A permanent disconnection between source and drain of CMOS transistor is modeled as *stuck-open fault.*
- In this faulty condition, a combinational circuit may behave as a sequential machine.



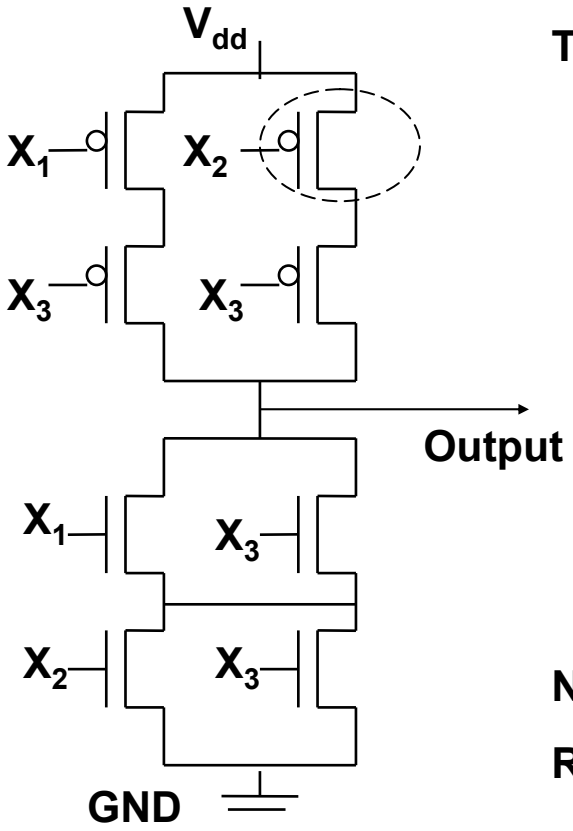
Test

Initialization vector (x_1x_2 : 11)

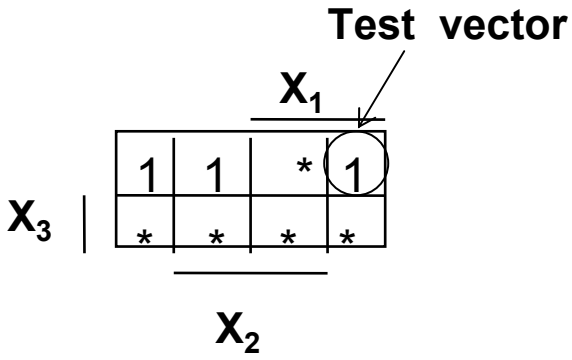
Test vector (x_1x_2 : 10)

Stuck-open Fault in CMOS NAND gate

Stuck-open Fault : Robust Testability



Testing requires two-pattern tests



Initialization vector: *

Non-robust test: (x_3,x_2,x_1) = (100, 001)

Robust test: (x_3,x_2,x_1) = (101,001), (011, 001)

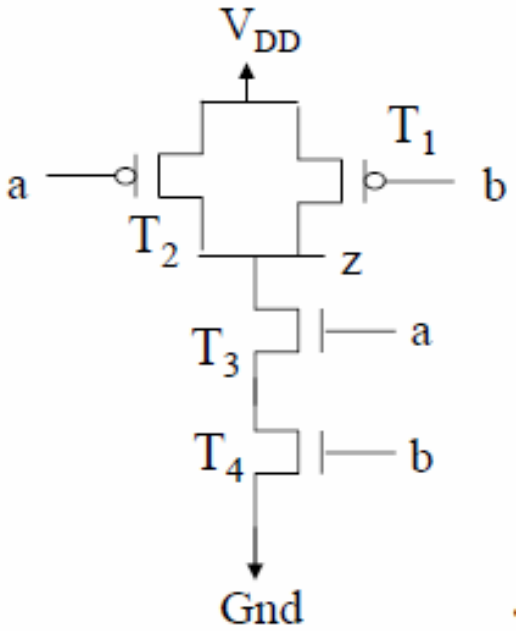
For robust testability:

Initialization and test vector must differ in single bit

: Single-Input-Change (SIC) pair

Stuck-short (on) Fault

- Stuck-short faults cause conducting path from V_{DD} to V_{SS}
  - Can be detected by monitoring steady-state power supply current I_{DDQ}
  - When input is low, both P and N transistors are conducting causing increased quiescent current, called IDDQ fault



- T1 stuck-on: (1,1) possible test pattern Output for T2, T2, T3 turned on should be 1
- T4 stuck-on: (1,0) possible test pattern Output for T2, T2, T3 turned on should be 0

$\Rightarrow$ Only one of these faults can be detected

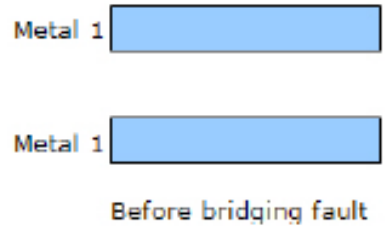
- Limitation of voltage monitoring techniques
- Current monitoring techniques needed

Geometric Fault Models

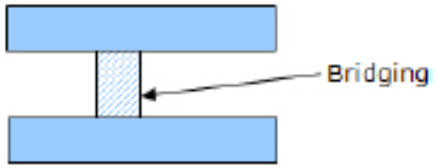
Bridging fault Models

- Derived from circuit layout
- Tests for resistive shorts between two normally unconnected nets
- Closest fault-model to real defects
- Bridging-faults are caused by manufacturing defects due to
  - Improper masking or etching
  - Loose or excess bare wires
  - Defective printed circuit boards
  - Shorting of pins of a chip

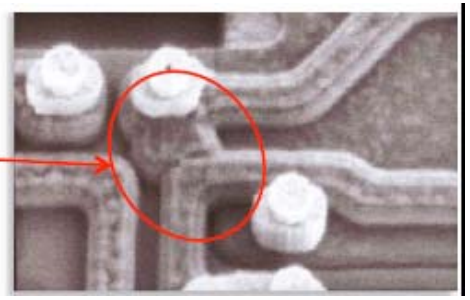
Bridging Fault Models

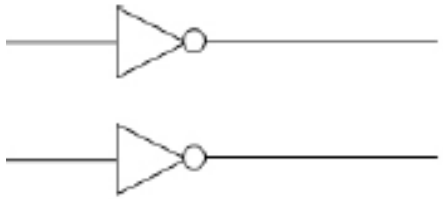


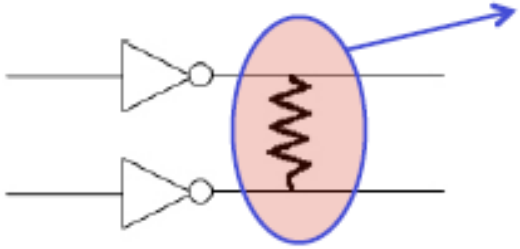
Before bridging fault



After bridging fault



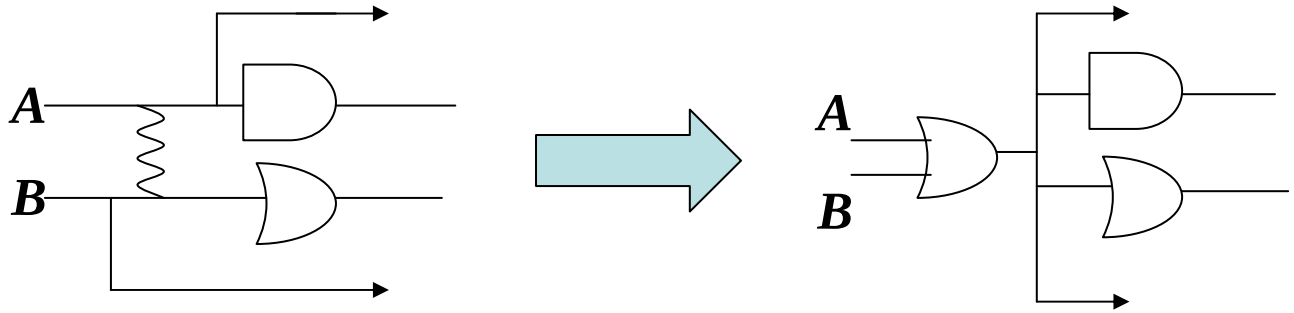




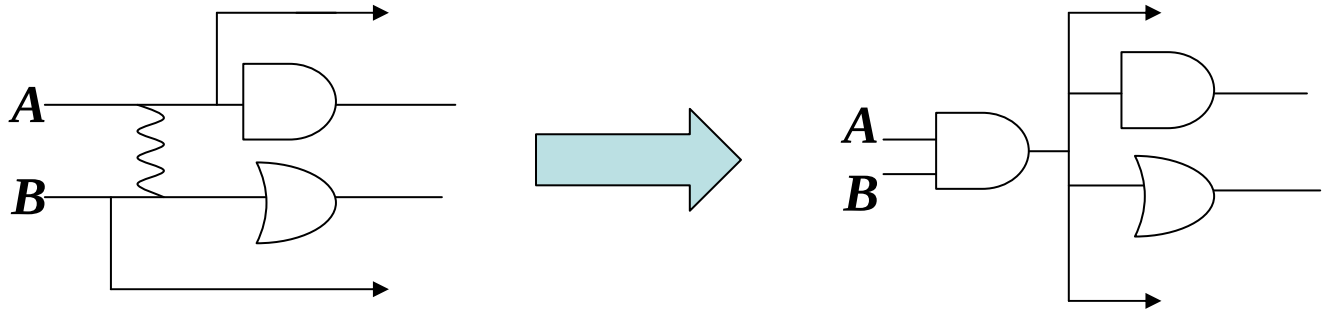
modelled as wired-AND, wired-OR or resistive shorts

Bridging Fault Models

OR-bridging

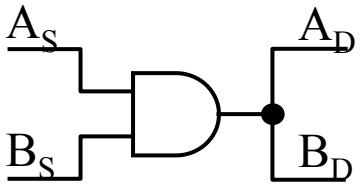


AND-bridging

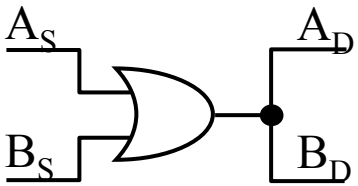


Bridging Fault Models

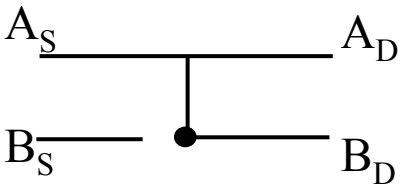
- Three different models
- Wired-AND/OR
- Dominant
- Dominant-AND/OR



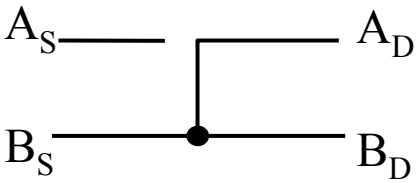
Wired-AND



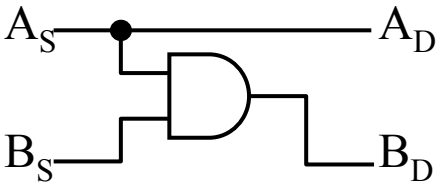
Wired-OR



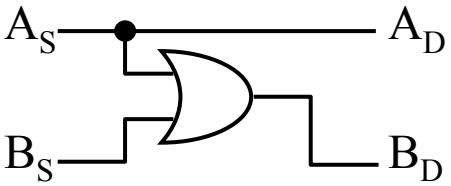
A dominates B



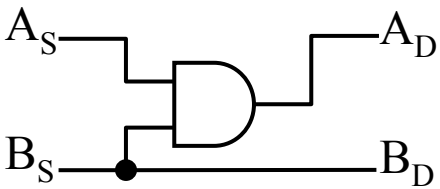
B dominates A



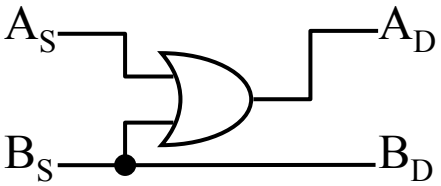
A dominant-AND B



A dominant-OR B



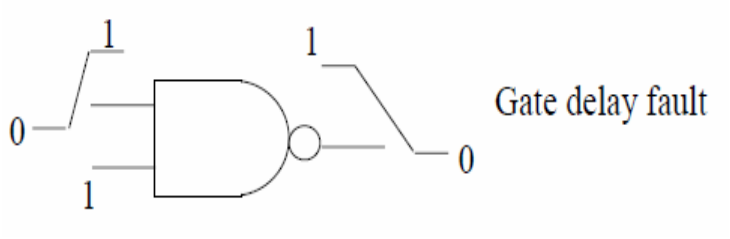
B dominant-AND A



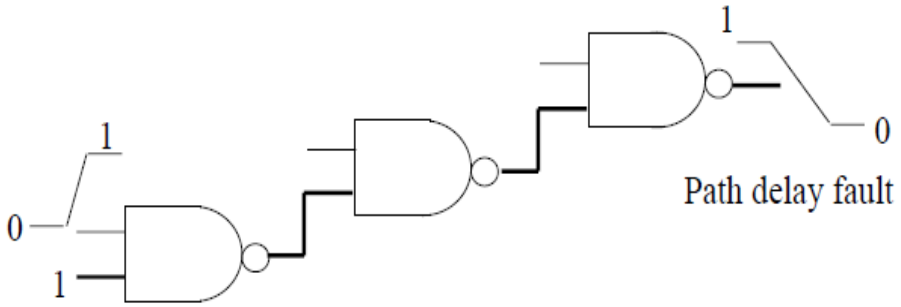
B dominant-OR A

Delay Fault Models

- Failures that cause logic circuits to malfunction at the desired clock rate are modeled as delay faults
- Affect propagation delay of the circuit, circuit fails at high speeds
- More important for high-speed circuits
- Gate delay fault (GDF): slow 1-to-0 or 0-to-1 transition at a gate output
- Path delay fault (PDF): exists a path from a primary input to primary output that is slow to propagate a 0-to-1 or 1-to-0 transition



Gate delay fault

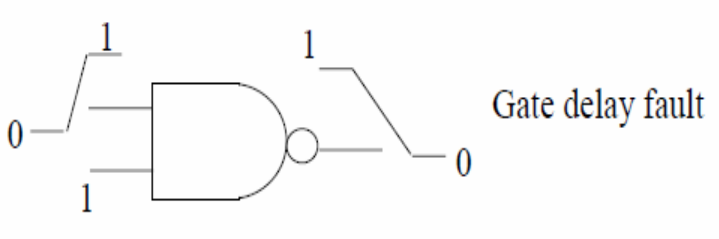


Path delay fault

Delay Fault Models

Transition faults:

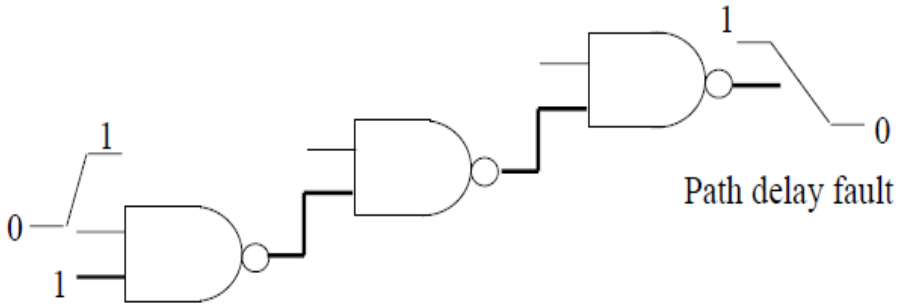
- Two faults per gate; slow-to-rise and slow-to-fall.
- Tests are similar to stuck-at fault tests. For example, a line is initialized to 0 and then tested for s-a-0 fault to detect slow-to-rise transition fault.
- Models spot (or gross) delay defects.



Gate delay fault
(transition-delay fault)

Path-Delay Faults:

- Two PDFs (rising and falling transitions) for each physical path.
- Total number of paths is an exponential function of gates. Critical paths, identified by static timing analysis (e.g., *Primetime* from Synopsys), must be tested.

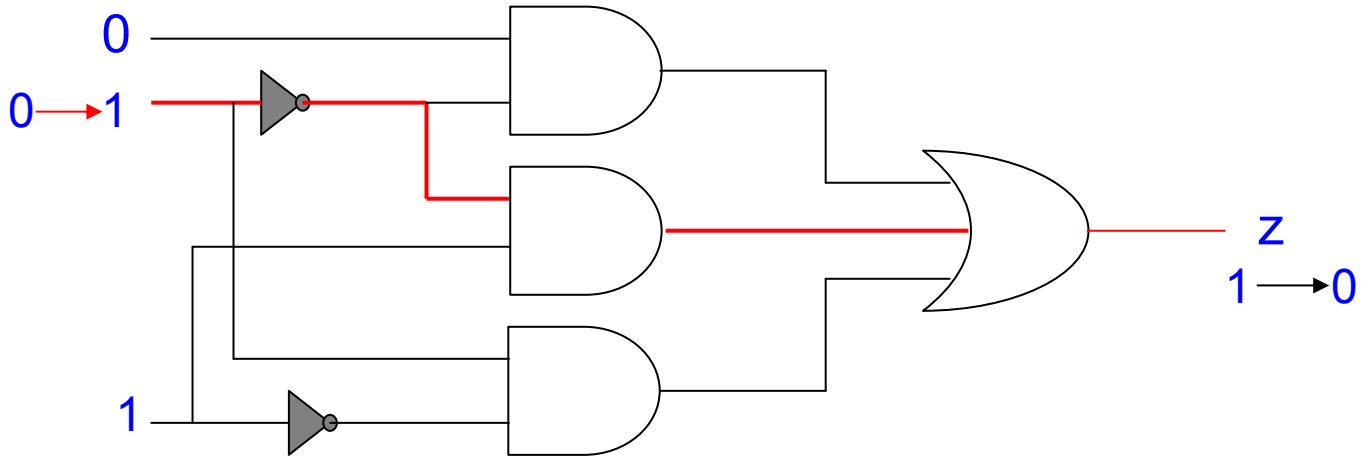


Delay Fault Terminology

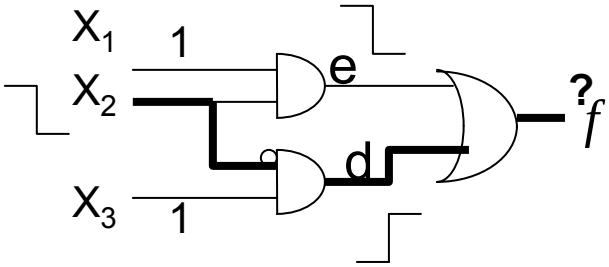
- Two-pattern tests
- Test is *robust* if independent of delays in the rest of the circuit, else *non-robust*
- **Robust Test**
- A robust test guarantees the detection of a delay fault of the target path, irrespective of delay faults on other paths.
- A robust test is a combinational vector-pair, V1, V2. V1 is called the initialization vector and V2 is called the test vector.

Path Delay Fault

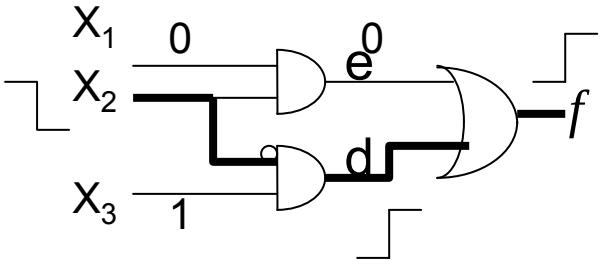
- A circuit is said to have a path-delay fault, if the total delay along some path exceeds the system clock interval
- Detection of a path-delay fault requires a two-pattern test



Path Delay Fault (Robust and Non-robust Test)



(111,101) : non-robust test



(011,001) : robust test

- A test is robust

  if it is not invalidated by delays in other paths

- A circuit is called delay testable if every path delay fault in it has a robust test

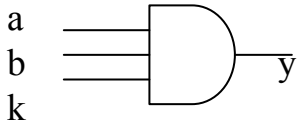
Transition Delay Faults

Two kinds of transition faults:

slow-to-rise and *slow-to-fall.*

Slow-to-rise (fall) transition fault temporarily behaves like a stuck-at-0 (1) fault.

Testing of such faults requires two-pattern tests, each consisting of an initialization and a test vector.



	t_1	t_2	t_3	t_4	t_5	t_6	t_7
a	1	0	1	1	1	1	1
b	1	1	1	0	1	1	1
k	1	1	1	1	1	0	1

VLSI Test Technology:Practical ATPG

- **Automatic Test Pattern Generation (ATPG)**
  - Algorithms generating sequence of test vectors for a given circuit based on specific fault models

Important ATPG algorithms

- D algorithm (Roth, 1966)
- PODEM (Goel, 1981)

  (Path Oriented Decision Making (PODEM) Algorithm)
- FAN (Fujiwara and Shimino, 1983)
- SOCRATES (Schulz et al., 1988)
- EST (Giraldi and Bushnell., 1990)
- Recursive Learning (Kunz and Pradhan., 1992)

Automatic Test Equipment (ATE)

- Test application is performed by *automatic test equipment (ATE)*
- **ATE consists of**
  - Computer – for central control and flexible test & measurement for different products
  - Pin electronics & fixtures – to apply test patterns to pins & sample responses
  - Test program – controls timing of test patterns & compares response to known good responses

Automatic Test Equipment (ATE)

Agilent 93000 series

Fault Simulation

- **Fault simulation**
  - Emulates fault models in CUT and applies test vectors to determine fault coverage
  - Simulation time (significant due to large number of faults to emulate) can be reduced by
    - Parallel, deductive, and concurrent fault simulation

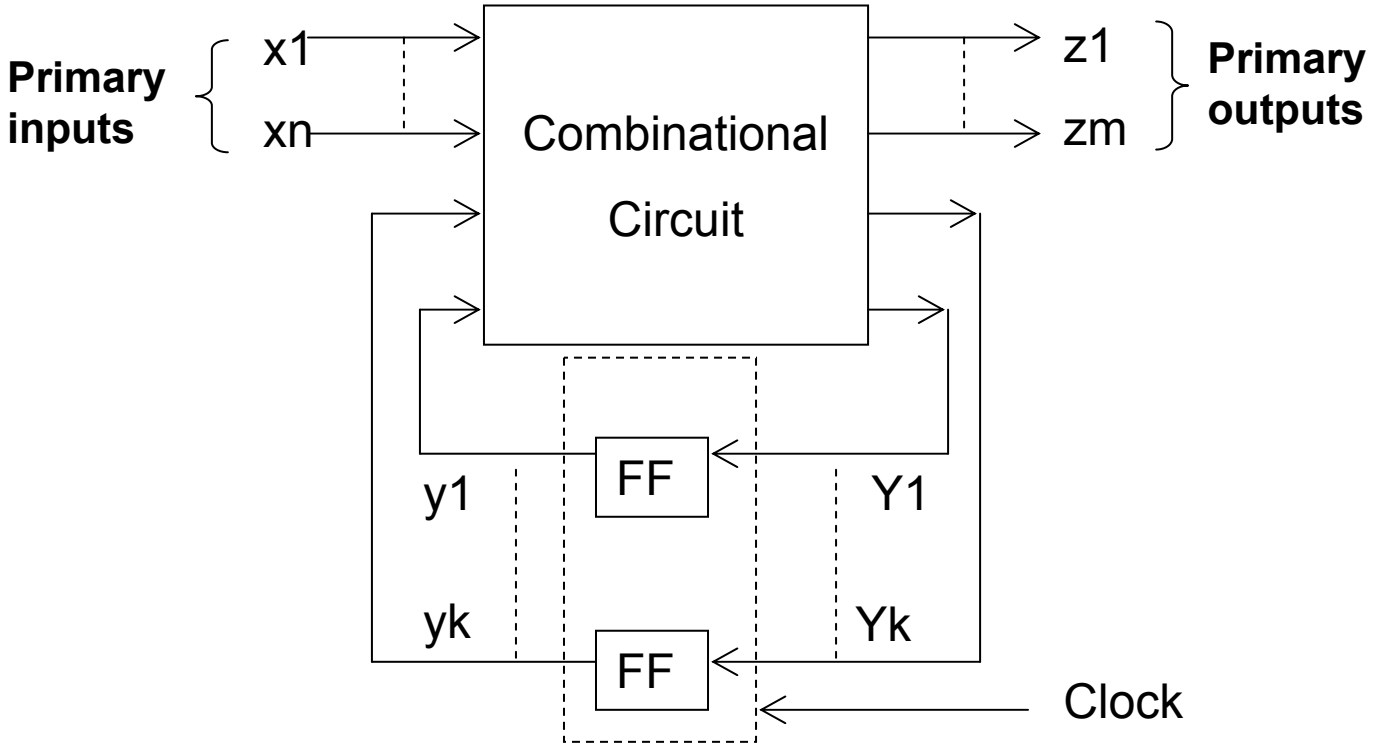
Test Generation Problem for Combinational Circuits

- For a completely fan-out free circuit, every single stuck-at fault is detectable
- If the circuit is in two-level, test generation problem is solvable in polynomial time
- For general circuit, the problem is NP-complete

Test Generation in Sequential Circuits: Hard Problems

- A sequential circuit has memory in addition to combinational logic.
- Test for a fault in a sequential circuit is a sequence of vectors, which
  - Initializes the circuit to a known state
  - Activates the fault, and
  - Propagates the fault effect to a primary output
- Methods of sequential circuit ATPG

Testing Sequential Circuits: Hard Problems



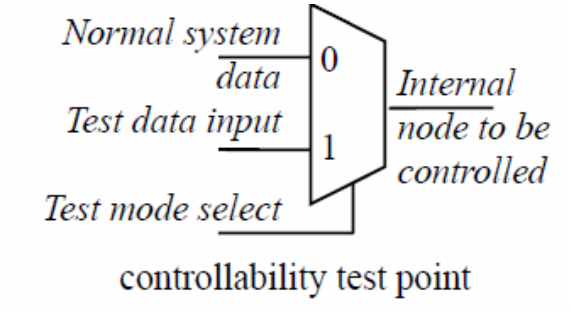
General model of a sequential circuit

Design for Testability (DFT)

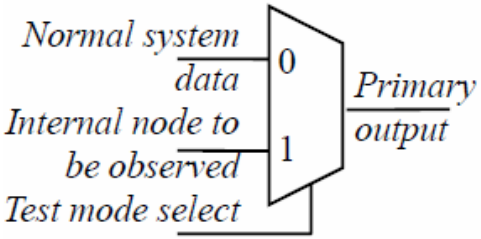
- **Design for Testability (DFT)**
  - Generally incorporated in design
  - Goal: improve controllability and/or observability of internal nodes of a chip or PCB
- **Three Basic Approaches**
  - Ad-hoc techniques
  - Scan design
    - Partial Scan
    - Boundary Scan
  - Built-In Self-Test (BIST)
- Design the circuit in such a manner that testing becomes fast, with high fault coverage and economical
- Integration of Design and Test is referred to as DFT.
- DFT refers to hardware design styles or added hardware that reduces test generation complexity.

Design for Testability (DFT)

- Ad-hoc DFT techniques
  - Add internal test points (usually multiplexers) for
    - Controllability
    - Observability
  - Added on a case-by-case basis
    - Primarily targets “hard to test” portions of chip



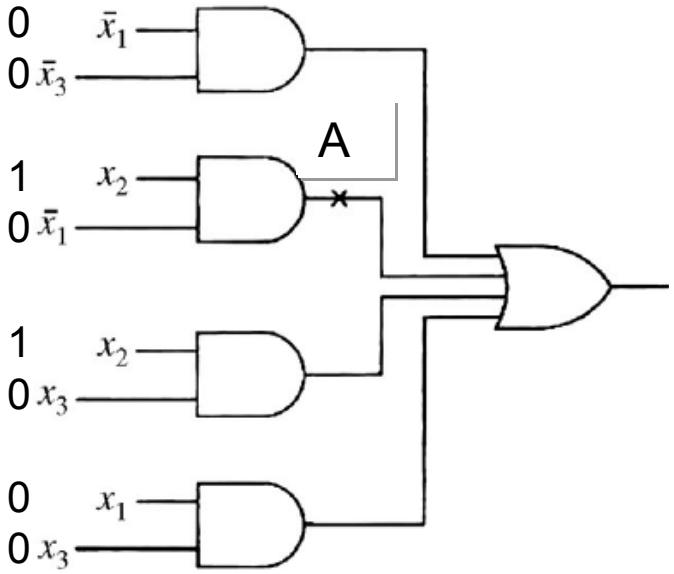
controllability test point



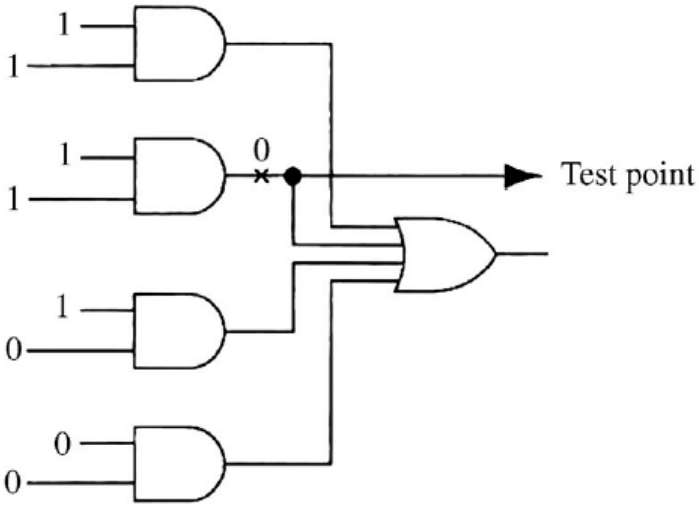
observability test point

Design for Testability (DFT)

Example: Ad-hoc DFT techniques



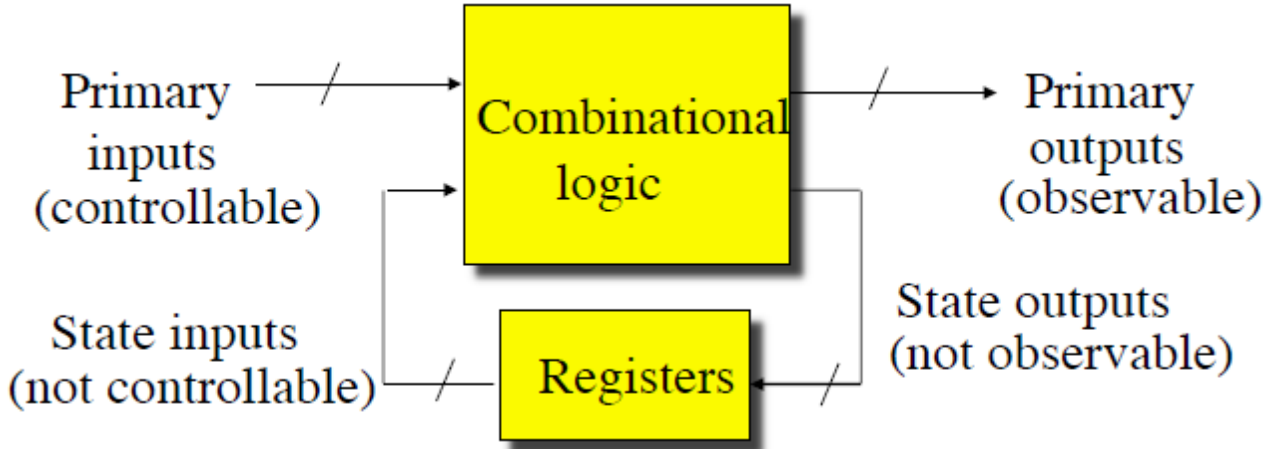
Circuit with undetectable fault A s-a-0



Fault detected when x1x2x3=010 is applied

Testing Sequential Circuits: Hard Problems

- Difficult problem-internal states cannot be directly controlled and observed
- Long test sequences are necessary
- Poor initializability, controllability and observability of memory elements
- Cyclic structure of the circuit is mainly responsible for the test generation complexity

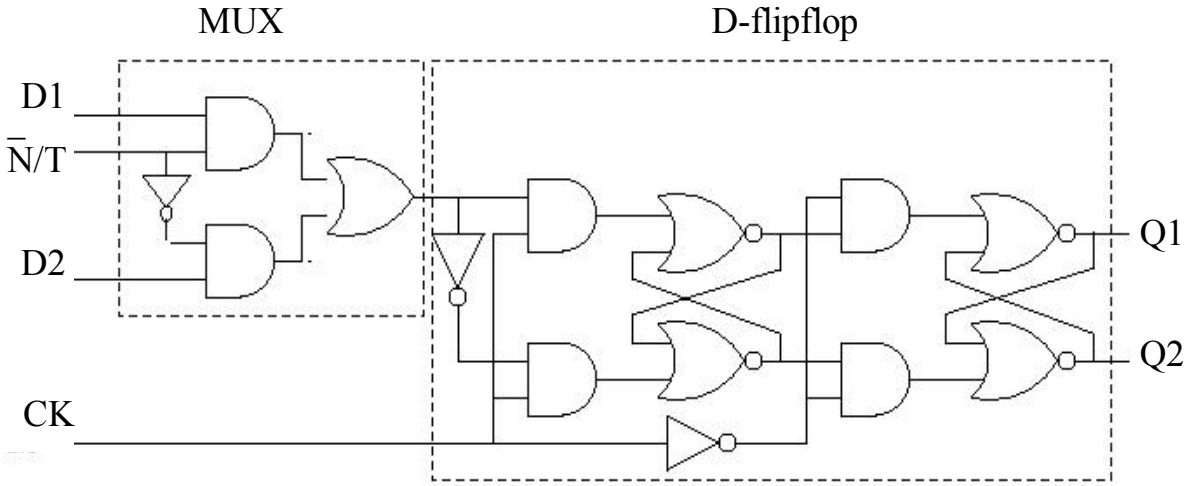


Scan Design

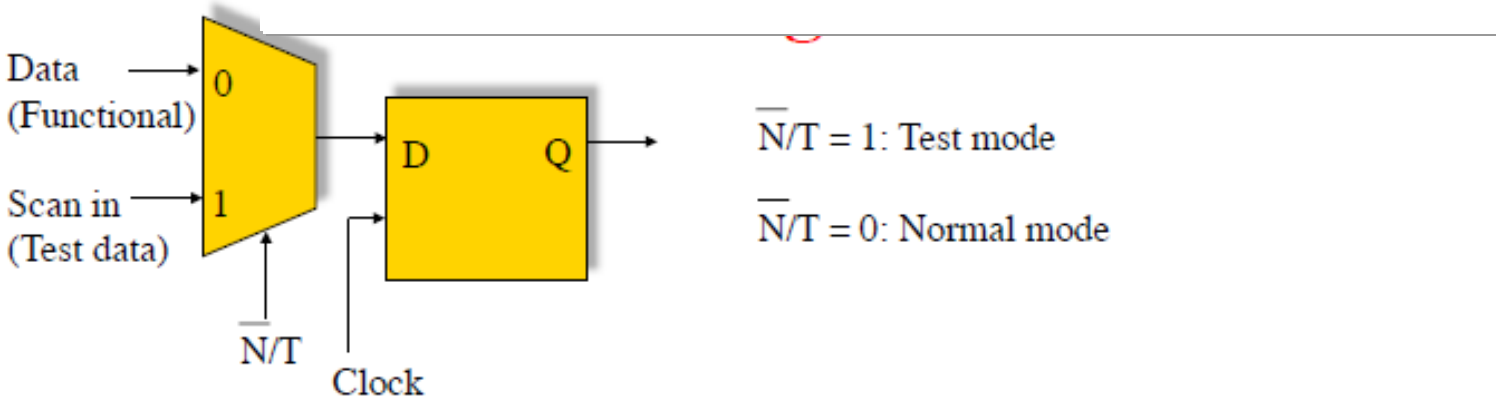
- Scan Design: is a design technique that makes it easy to test a *sequential circuit* by replacing the flip-flops with scan registers.
  - Basic units of scan design
    - Scan Register (shift register during testing)
    - MUX

Scan Design

$\overline{N}$/T: low, test mode, test vectors shifted in through D2
$\overline{N}$/T: high, normal mode, data are shifted in through D1

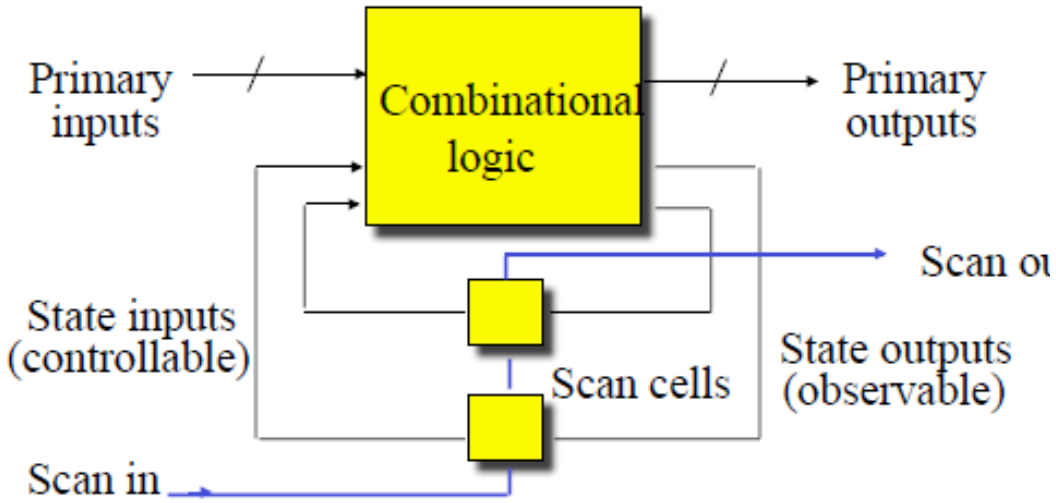


Typical Scan register



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

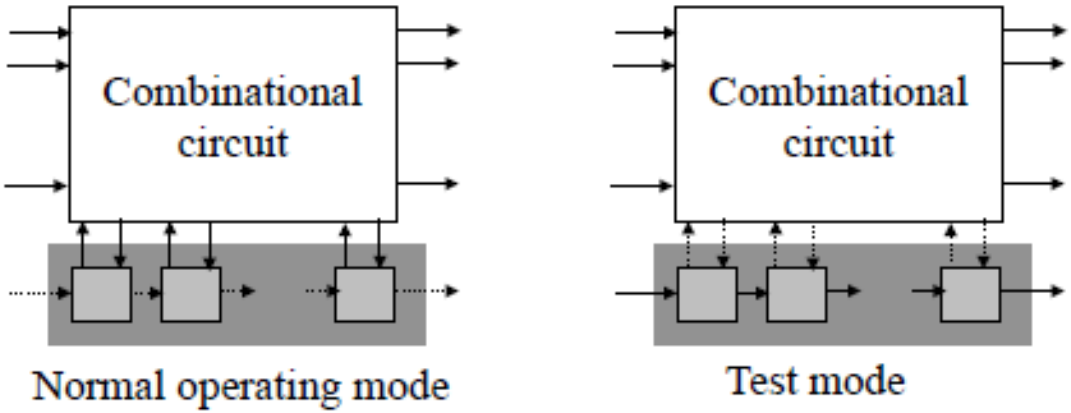


Scan Design

- 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

- 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 Design

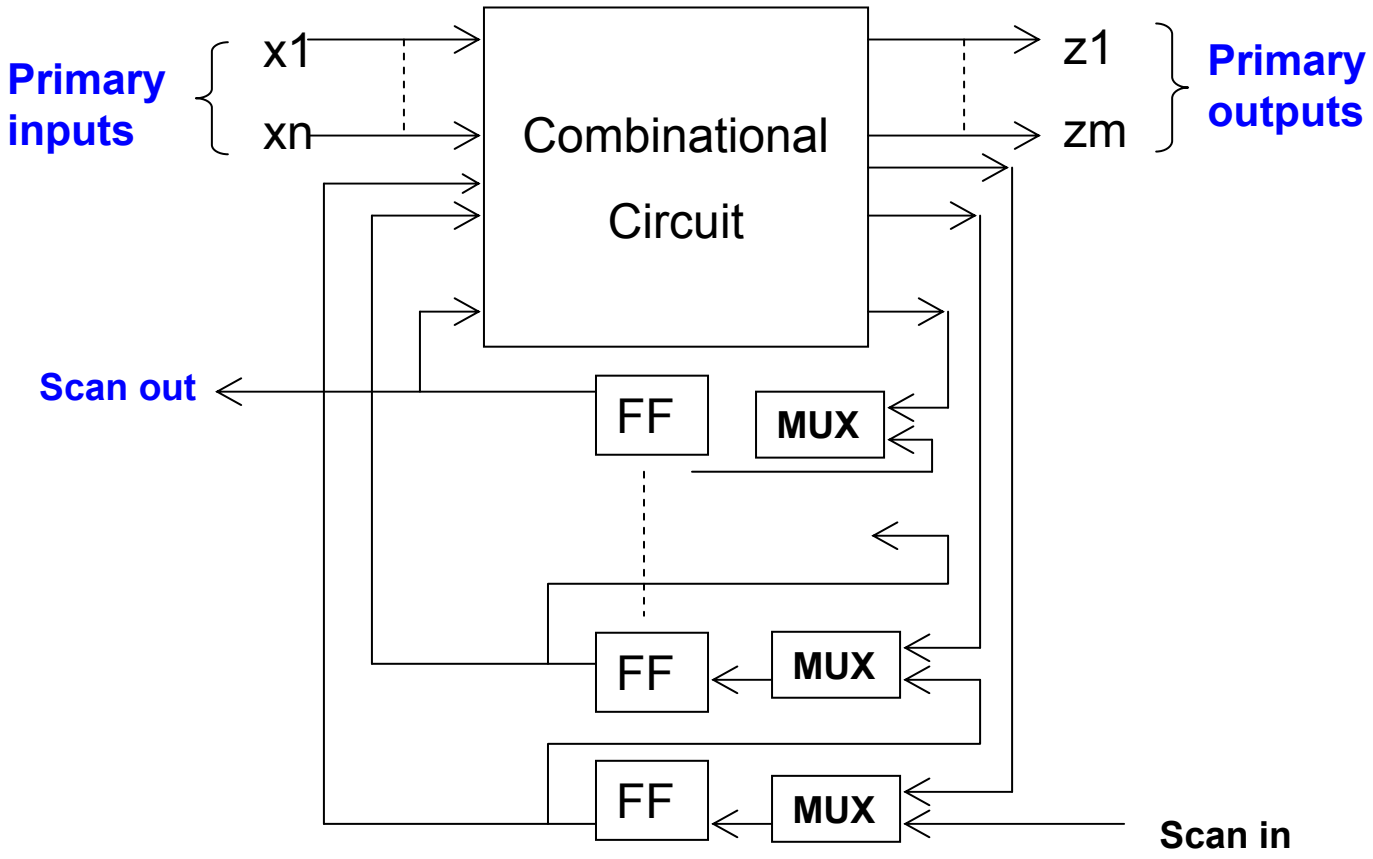
Comments on Scan Design

- 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.

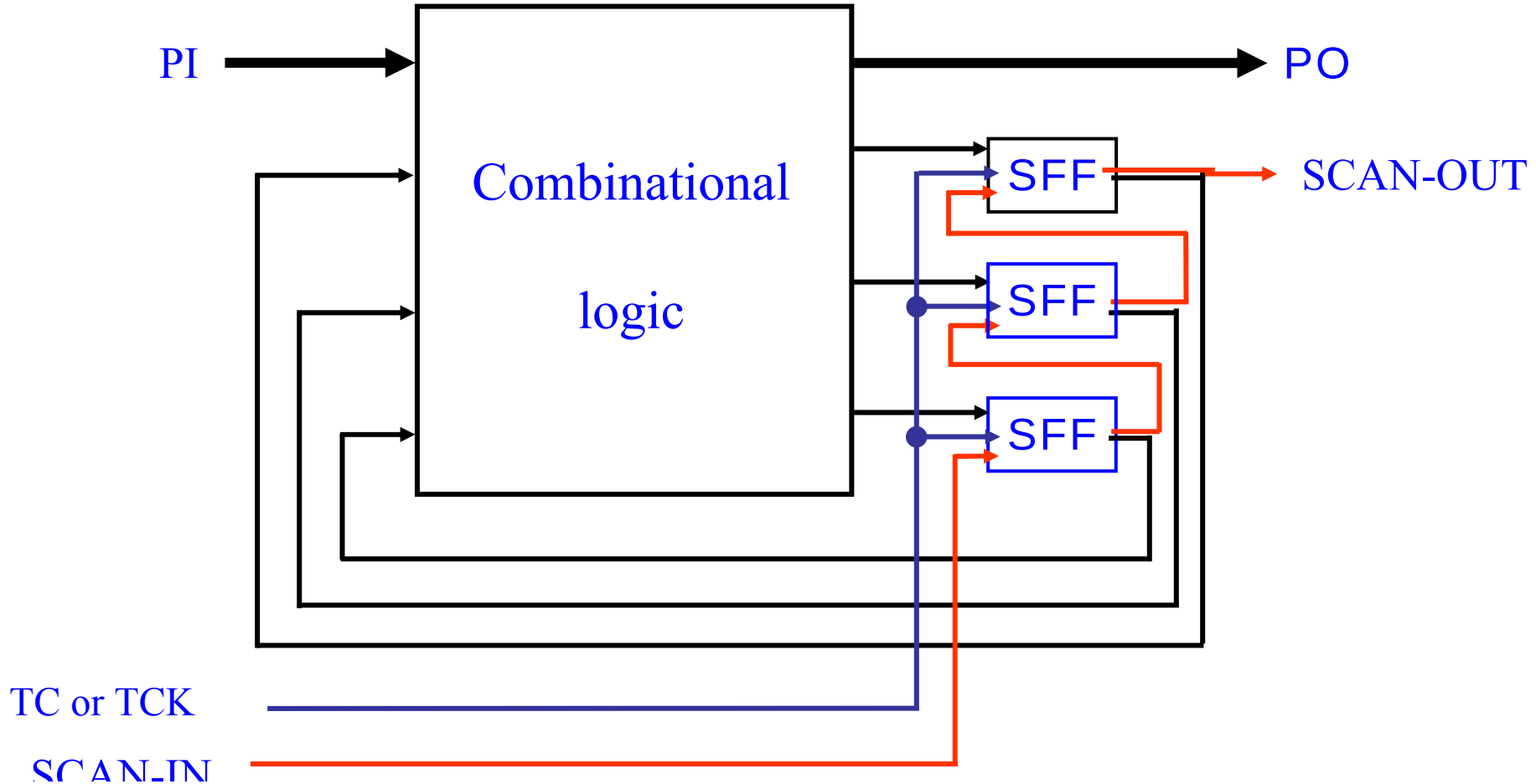
Scan Design



Full scan technique

Adding Scan Structure

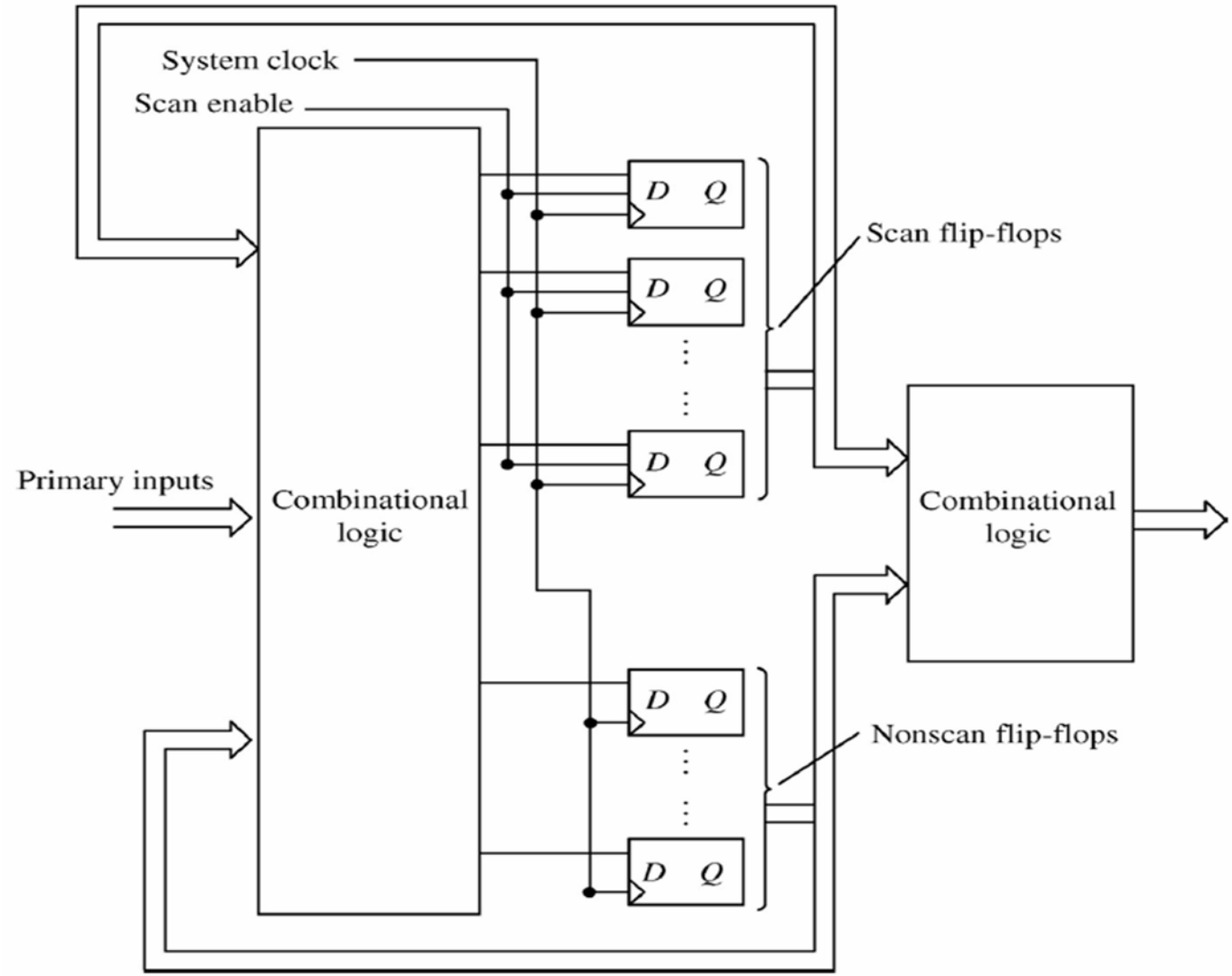
- Scan flip-flops can be distributed among any number of shift registers, each having a separate *scanin* and *scanout* pin.
- Test sequence length is determined by the longest scan shift register.
- Just one *test control* (TC) pin is essential.



Partial Scan

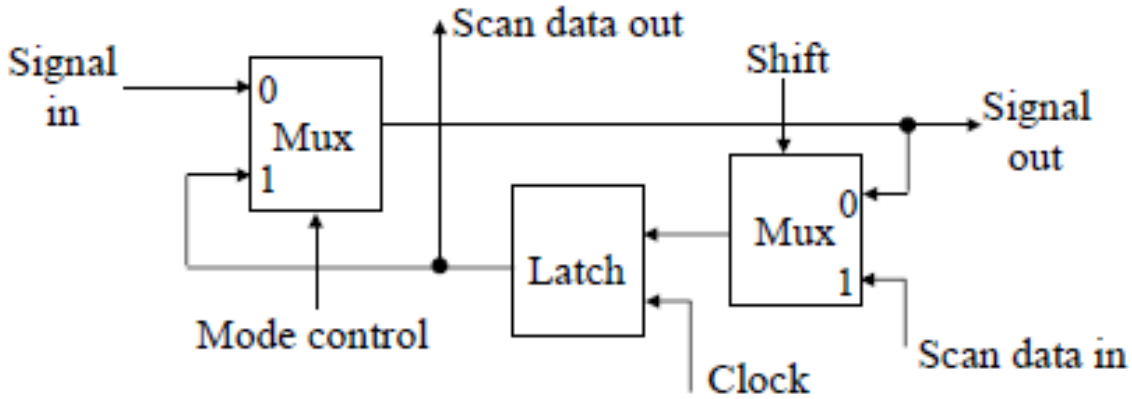
- **Basic idea**
  - Select a carefully chosen subset of FFs for scan
- Advantages
  - Lower overhead in area and speed

Partial Scan

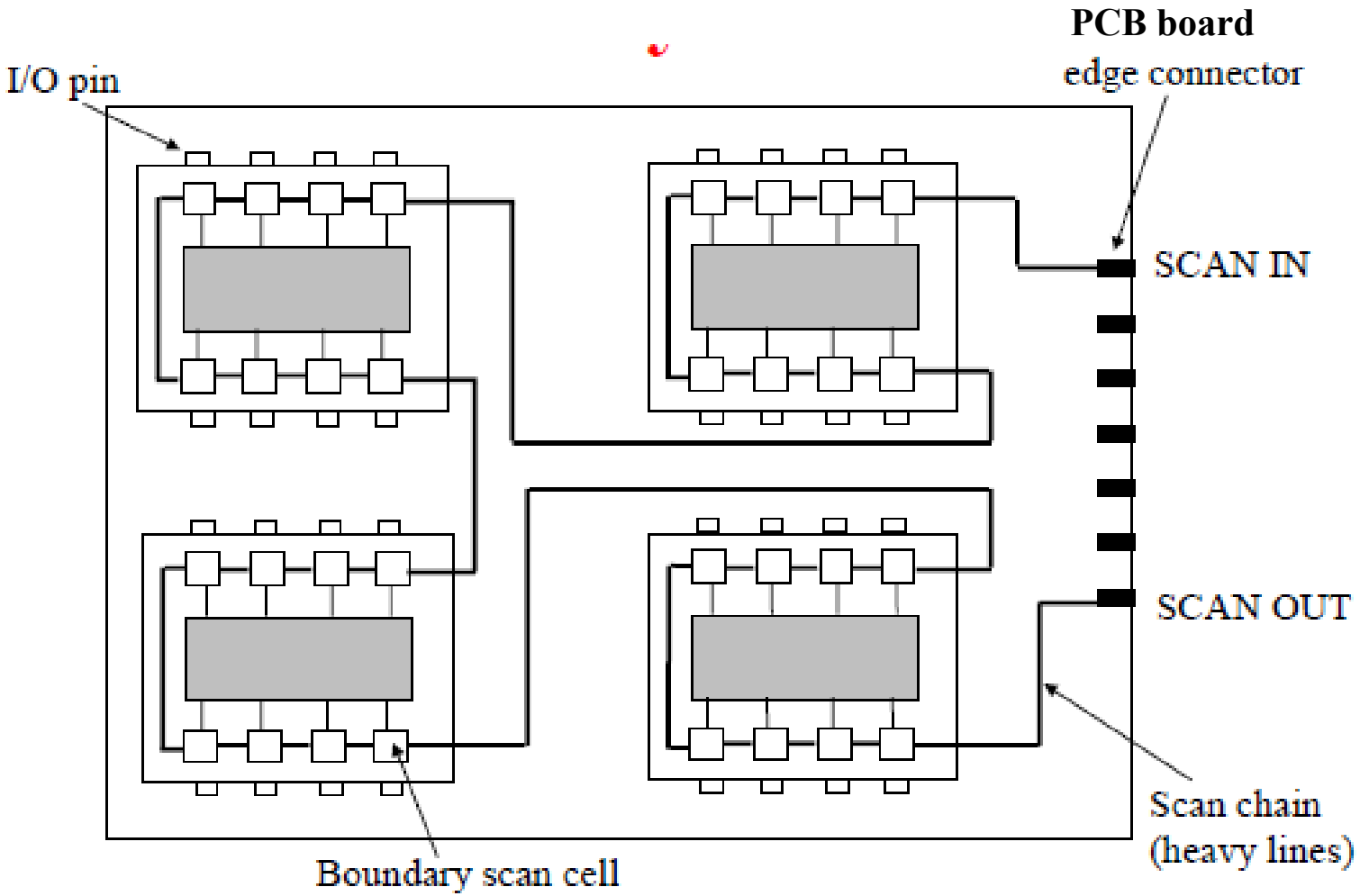


Boundary Scan

- 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 core-based systems-on-a-chip

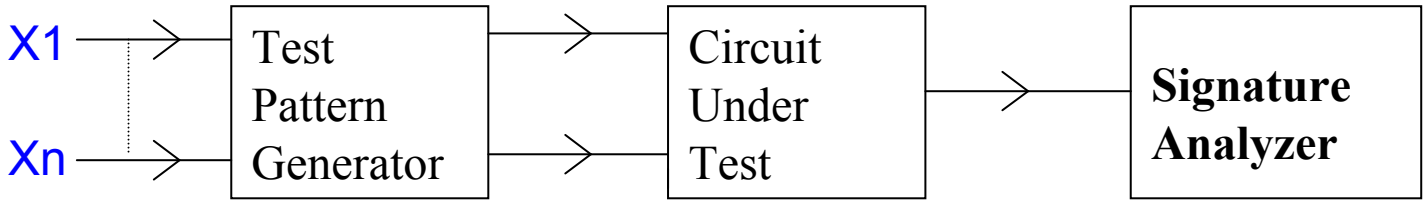


Boundary Scan



Built-in-self-Test (BIST)

- It is a design technique in which portion of a circuit is used to test the circuit itself



General Structure of a BIST technique

Built-in-self-Test (BIST)

Test pattern generator (TPG)
-generates and applies a sequence of test vectors to the circuit under test (CUT).

Exhaustive pattern – 2^n
pseudo exhaustive pattern: 2^W where w is a set of n.

Random patterns: generates random test patterns.

Signature analyzer: The output responses of CUT are collected and compressed by signature analyzer (SA).

Built-in-self-Test (BIST)

BIST Motivation

- Useful for field test and diagnosis (less expensive than a local automatic test equipment)
- Software tests for field test and diagnosis
- Hardware BIST benefits:
- Lower system test effort
- Improved system maintenance and repair
- Improved component repair
- Better diagnosis

Built-in-self-Test (BIST)

- **BIST Benefits**
- Single combinational / sequential stuck-at faults
- Reduced testing and maintenance cost
- Lower test generation cost
- Reduced storage / maintenance of test patterns
- Simpler and less expensive ATE
- Can test many units in parallel
- Shorter test application times
- Can test at functional system speed

Built-in-self-Test (BIST)

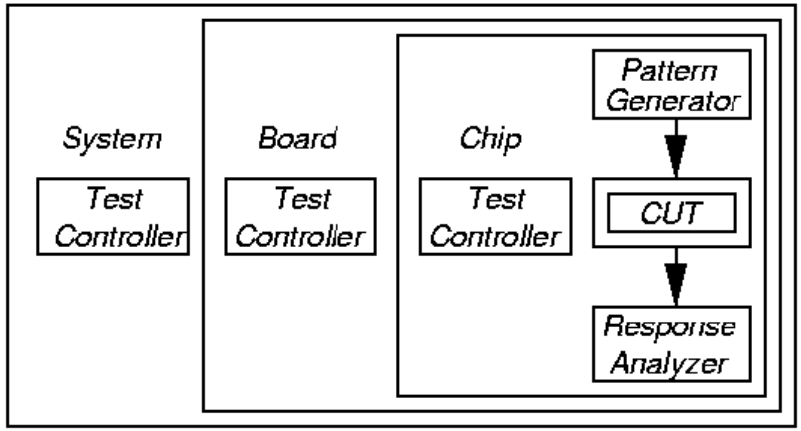
- **Some Definitions**
- BILBO – Built-in logic block observer, extra hardware added to flip-flops so they can be reconfigured as an LFSR pattern generator or response compacter, a scan chain, or as flip-flops
- Concurrent testing – Testing process that detects faults during normal system operation
- CUT – Circuit-under-test
- Exhaustive testing – Apply all possible 2^n patterns to a circuit with n inputs
- Irreducible polynomial – Boolean polynomial that cannot be factored
- LFSR – Linear feedback shift register, hardware that generates pseudo-random pattern sequence

Built-in-self-Test (BIST)

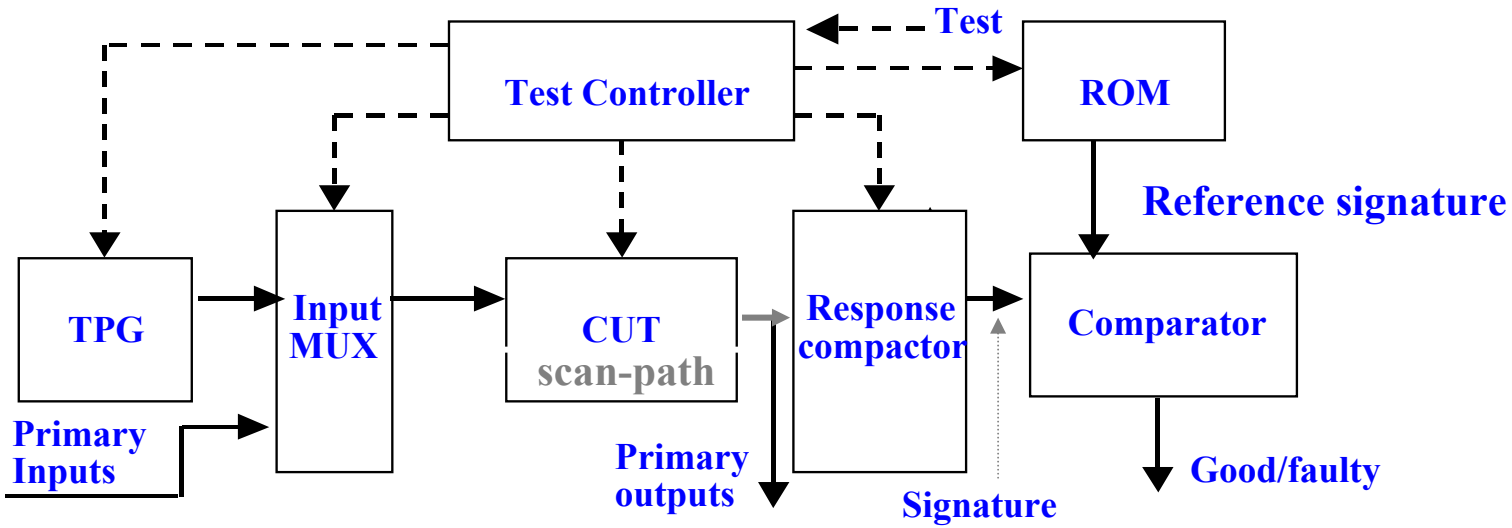
- **More Definitions**
- *Primitive polynomial*
- *Pseudo-exhaustive testing* – Break circuit into small, overlapping blocks and test each exhaustively
- *Pseudo-random testing* – Algorithmic pattern generator that produces a subset of all possible tests with most of the properties of randomly generated patterns
- *Signature* – Any statistical circuit property distinguishing between bad and good circuits
- TPG – Hardware *test pattern generator*

Built-in-self-Test (BIST)

- **BIST Process**
- Test controller – Hardware that activates self-test simultaneously on all PCBs
- Each board controller activates parallel chip
- BIST Diagnosis effective only if very high fault coverage



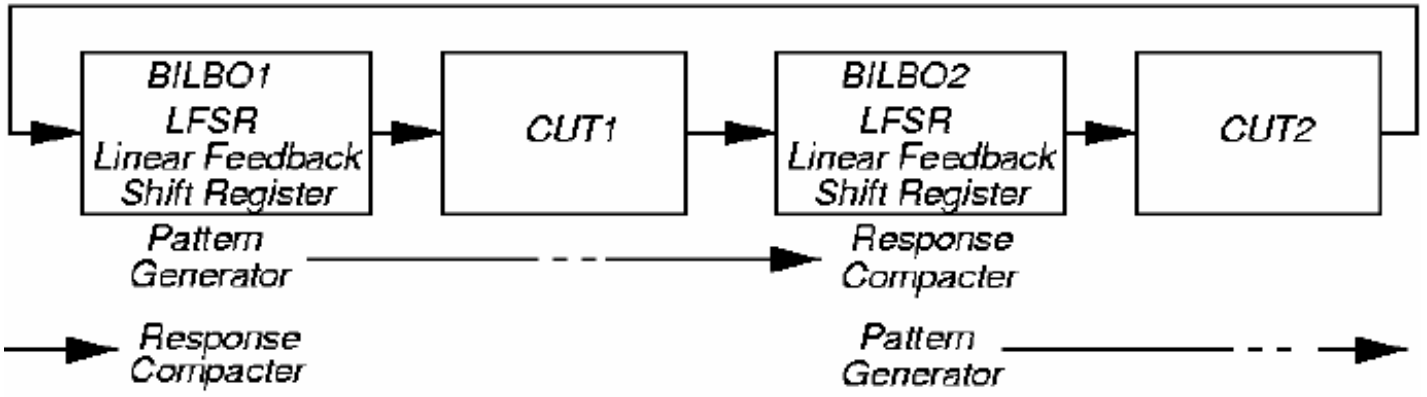
BIST Architecture



Block diagram of BIST

- Note: BIST cannot test wires and transistors:
  - From PI pins to Input MUX
  - From POs to output pins

BILBO – Works as Both a PG and a RC



- *Built-in Logic Block Observer* (BILBO) -- 4 modes:
  - Flip-flop
  - LFSR pattern generator
  - LFSR response compacter
  - Scan chain for flip-flops

Pattern Generation

- Store in ROM – too expensive
- Exhaustive
- Pseudo-exhaustive
- Pseudo-random (LFSR) – Preferred method
- Binary counters – use more hardware than LFSR
- Modified counters
- Test pattern augmentation
  - LFSR combined with a few patterns in ROM

Test Challenges in Nanometer Era

- Shrinkage of feature size has made a dramatic impact on test.

- *System-on-chip* (SOC) designs embed billion of transistors running in the gigahertz range. These designs can include all varieties of *digital, analog, mixed-signal, memory, optical*, ***micro-electromechanical system*** (MEMS), ***field programmable gate array*** (FPGA), and ***radiofrequency*** (RF) circuits.

- Testing for this complex design will be a significant challenge.

- *Scan and* BIST will not be sufficient to address the testing problems in the nanometer design according to *International Test Technology Roadmap (ITTR)*.

Test Challenges in Nanometer Era

- *According to ITTR* 2004, promising test techniques are required to deal with highly complex nanometer designs

- **Crosstalk noise** between adjacent interconnects due to capacitive and inductive coupling causes **signal integrity** problem which is extremely difficult to detect

- P**ower integrity** problem will be considered.

- **Process variation** problem can make delay testing extremely complex [Wang 2004].

- New fault model called ***drowsy fault that*** causes a memory cell to fall asleep forever.

Test Challenges in Nanometer Era

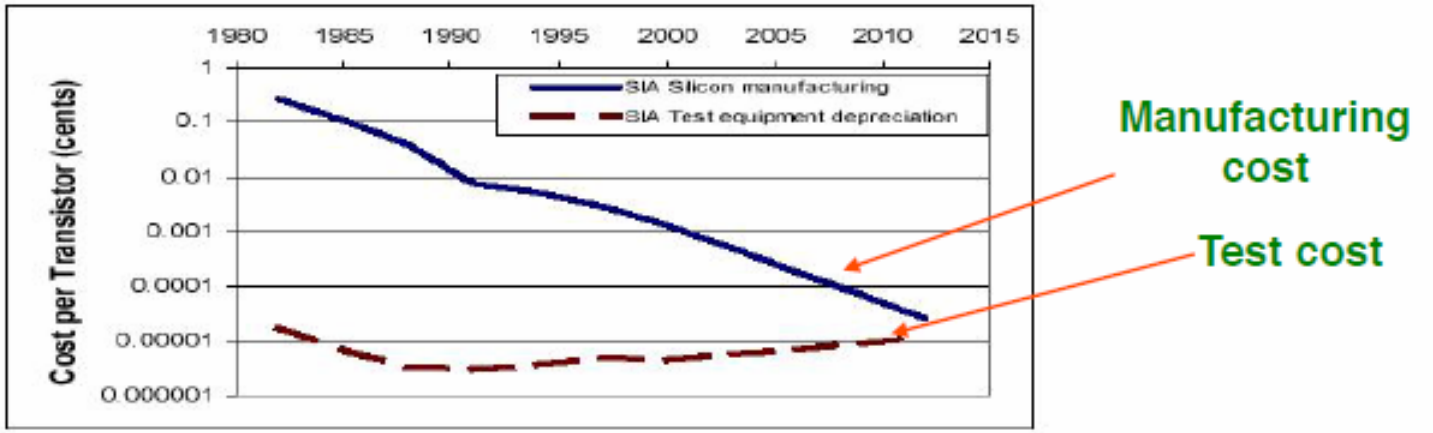
- Nanometer designs with feature size <45nm beyond 2010 include the device under test (DUT) to automatic test equipment (ATE) interface, test methodologies, defect analysis, failure analysis, and disruptive device technologies.

Test challenges for nanometer designs

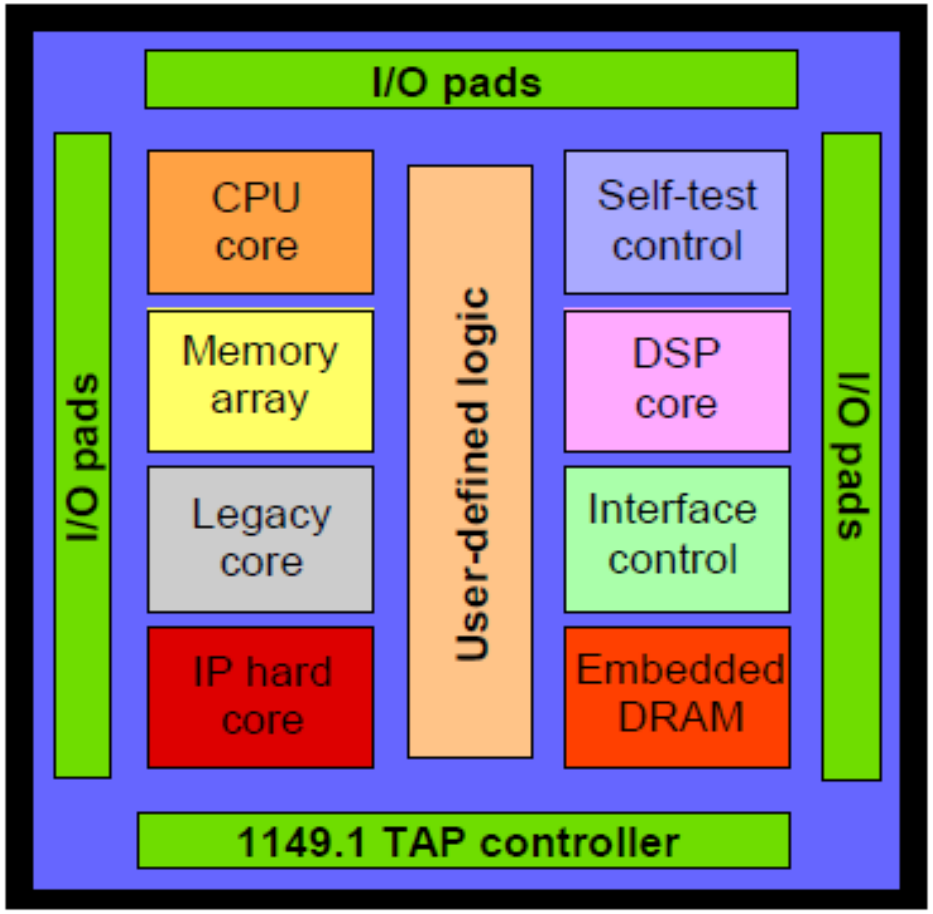
- Development of new DFT and DFM methods for digital circuits, analog Mixed Signal circuits, MEMS, and Sensors
- Development of the means to reduce manufacturing test costs as well as enhance device reliability and yield
- Development of techniques to facilitate defect analysis and failure analysis.

System-on-Chip (SOC) Testing: Motivation

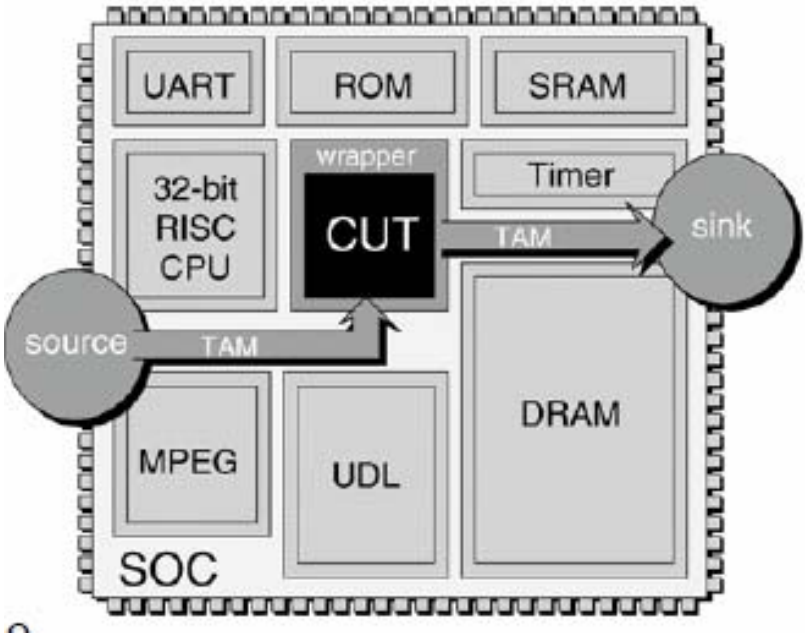
- System-on-chip (SOC) integrated circuits based on embedded intellectual property (IP) cores are now Common place
  - SOCs include processors, memories, peripheral devices, IP cores, analog cores
  - Low cost, fast time-to-market, high performance, low power
  - Manufacturing test needed to detect manufacturing defects



System-on-Chip (SOC) Testing: Motivation



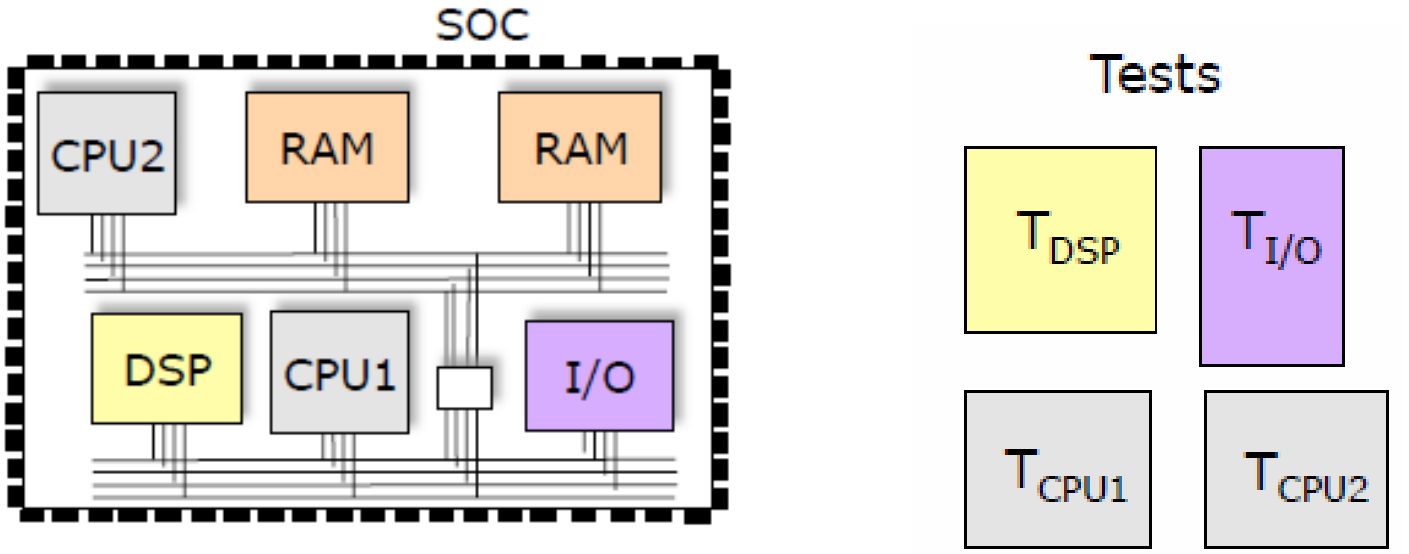
- Test access is limited
- Test sets must be transported to embedded logic



- Philips Nexperia™ PNX8550 SOC: 338,839 flip-flops, 274 embedded cores, 10M logic gates, 40M logic transistors!

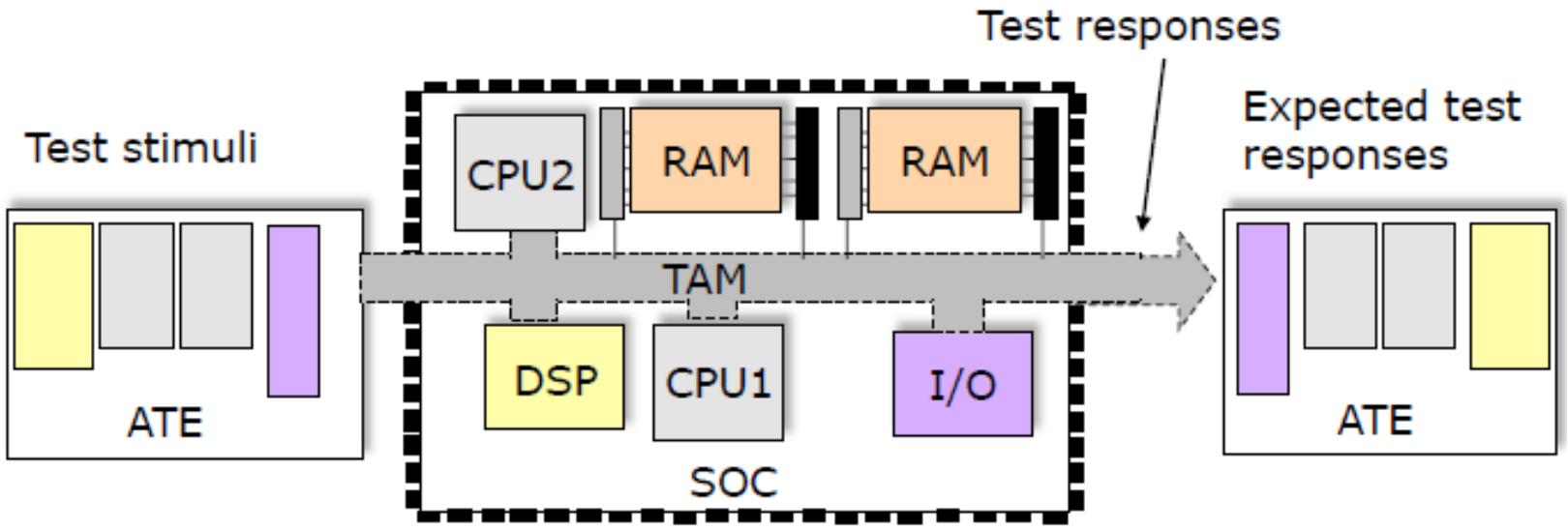
System-on-Chip (SOC) Testing: Motivation

- Shorten production cycles, and increasing complexity of modern electronic systems has forced designers to employ reuse based designs approaches.
- System-on-Chip (SOC) is an example of such reuse based design approach where pre-designed, pre-verified cores are integrated into a system.



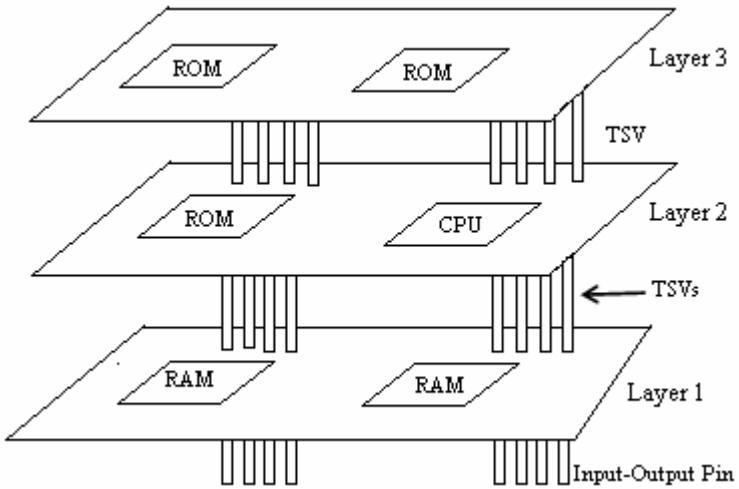
System-on-Chip (SOC) Testing

- Test access mechanism (TAM)
- An ATE is used to transport the test stimuli to the SOC. The produced responses are transported back to the ATE where they are compared with the expected responses.
- Memory cores are usually tested using a built-in self-test

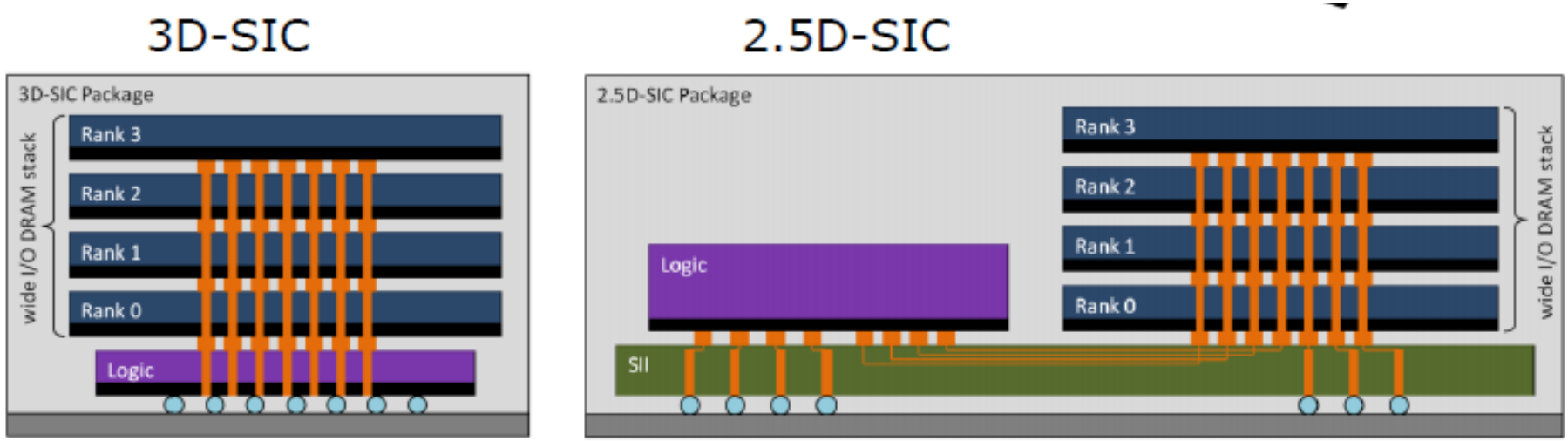


Testing of 2.5D/3D ICs

- Two or more layers of active electronic components are integrated both vertically into a single circuit.
- Advantage of 3D over 2D
- Lower interconnect
- Higher performance
- Higher packing density
- Implementation of mixed technology chips



Testing of 2.5D/3D ICs



Conclusions

- For digital logic single stuck-at fault model offers best advantage of tools and experience.
- Many other faults (bridging, stuck-open and multiple stuck-at) are largely covered by stuck-at fault tests.
- Stuck-short and delay faults and technology-dependent faults require special tests.
- Memory and analog circuits need other specialized fault models and tests.
- Test generation problem is NP complete
- Test Problem is intractable for large sequential circuits
- Complexity of test generation problem for combinational circuits seems to be polynomial
- An alternative approach is needed for testing large sequential circuits
- Future challenges are development of new DFT/DFM for Nanometer design

Thanks