EEM412 Course
Introduction to IC Design – II

Instructor
Dr. İsmail Enis Ungan

Need for Testing

- Not all die on a wafer operate correctly due to the IC manufacturing process.
- The aim of testing is to determine which dies are good.
- Testing a die (chip) can occur at and at malfunction detection cost of
  - wafer level (0.1$)
  - packaged chip level (1$)
  - board level (10$)
  - system level (100$)
  - in the field (1000$)
- Manufacturing cost is low when malfunctioning chip fault is detected at early level.
- Tests fall into two main categories: functionality and manufacturing.

Functionality Tests

- Verify the function of a chip as a whole
  - It is part of a design
  - It is checked that the circuit is functionally equivalent to the given specification which may be in a form of C language, HDL, truth-table etc.
  - Functional equivalence is verified by running a simulator at two design different domains and/or design abstraction levels
  - It involves timing tests also.

- Experience says that “If you don’t test it, it won’t work!”.

Manufacturing Tests

- Verify the physical operation of every gate in the chip.
  - It is done during chip fabrication so that functional correctness is assumed.
  - Typical manufacturing defects are
    - Layer to layer shorts. Shorts to VDD or VSS, shorts between to nodes.
    - Discontinuous wires. Floating inputs, disconnected outputs.
    - Thin-oxide shorts to substrate. Shorts to VDD or GND.
  - I/O integrity tests
    - Level tests: Noise margins for the TTL, ECL, CMOS etc. I/O pads.
    - Speed test
    - IDD test

Test Process

- A walk through the test process;
  - Write stimulus and apply to simulator,
  - Simulator output is the activity file,
  - Filter the activity file, i.e. Remove internal signals and keep only inputs and outputs,
  - Generate test program,
  - Prepare Device-Under-Test (DUT) board or probe card,
  - Connect DUT board to tester and run the test program,
  - Tester compares outputs with the test program. If difference is found then chip is marked as faulty,
  - Tester vary the supply voltage and timing, i.e. Schmooing process, then outputs a Schmoo plot, which shows the speed sensitivity w.r.t. supply voltage by pass and fail marks on the plot.

Manufacturing Test Principles

- Critical factor is to incorporate methods of testing circuits.
  - Consider a combinational logic with n inputs. A sequence of 2^n inputs (test vectors) are required to exhaustively test the logic. This may take years if n is large.
- Fault Models
  - In order to find out manufacturing defects, faults are needed to be modeled.
  - Single-stuck-at model. For stuck-at faults, faulty gate input is modeled as stuck-at-one or stuck-at-zero.
  - Other models include bridging fault (short circuit), open circuit faults, and multiple faults.
  - The goal is not to try to model the real fault, but to develop tests to detect the real faults.

Manufacturing Test Principles

- Observability
  - of an internal circuit node is the degree to which one can observe that node at the outputs (pins) of an IC.
- Controllability
  - of an internal circuit node is a measure of ease of setting that node to logic-1 or logic-0.
- Fault Coverage
  - is a measure of quality of the test vectors. It is expressed as the percentage of all possible faults that the test vectors will detect.
  - Its definition varies;
    - In Structured Domain, it is the percentage of interconnections actually tested to verify that they can be set to logic-1 and logic-0.
    - In Behavioral Domain, it is the percentage of possible addressing mode/operation code combinations tested.

Manufacturing Test Principles

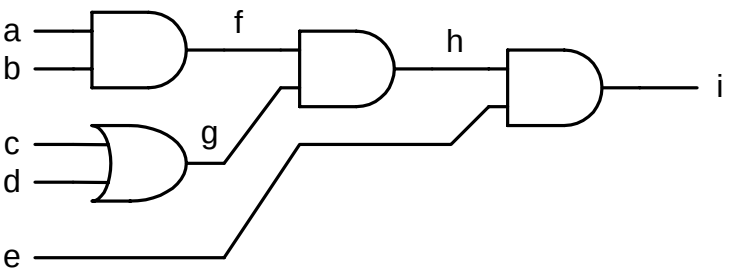
- Automatic Test Pattern Generation (ATPG)
  - A five-valued logic form is commonly used to implement test generation algorithms; 1, 0, X, D, D# (not D)
  - where D represents logic-1 in good machine/circuit, and logic-0 in a faulty one.
  - Propagation of five-valued logic through AND gate and OR gate is given by the truth tables as;

AND	B				
A	0	1	X	D	D#
0	0	0	0	0	0
1	0	1	X	D	D#
X	0	X	X	X	X
D	0	D	X	D	0
D#	0	D#	X	0	D#

OR	B				
A	0	1	X	D	D#
0	0	1	X	D	D#
1	1	1	1	1	1
X	X	1	X	X	X
D	D	1	X	D	1
D#	D#	1	X	1	D#

Manufacturing Test Principles

- Example: Application of five-valued logic;
- Primary inputs: **a**, **b**, **c**, **d**, **e**. Primary outputs: **i** . Gate under test: G3. Find an input vector in order to detect stuck-at fault at node **h**.



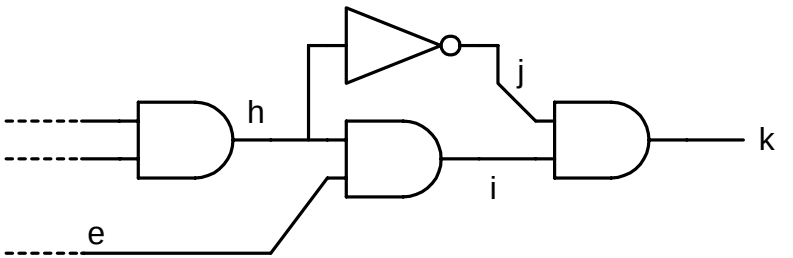
- Set h <= D. Propagate D to the primary output **i**. The path to primary output is called “sensitized path”.
- Set e <= 1, so that i <= D, hence **h** is observable.

Manufacturing Test Principles

- Set h to D via a set of primary inputs;
  - Find input vectors to test node **h** for Stuck-at-0 (SA0). By backtracking;
    - Set f <= 1 and g <= 1, so that h <= 1, i <= 1.
    - Set a <= 1 and b <= 1, so that f <=1.
    - Set (c <= 1 and d <= X) or (c <= X and d <= 1), so that g <= 1.
    - Then the Test Vector is {a,b,c,d,e} = {1,1,1,X,1} or {1,1,X,1,1}
  - Find input vectors to test node **h** for Stuck-at-1 (SA1). By backtracking;
    - Set (f <= 0 and g <= X) or (f <= X and g <= 0), so that h <= 0, i <= 0.
    - Set (a <= 0 and b <= X) or (a <= X and b <= 0), so that f <= 0.
    - Set a <= X and b <= X, so that f <= X.
    - Set c <= 0 and d <= 0, so that g <= 0.
    - Set c <= X and d <= X, so that g <= X.
    - Then the Test Vector is {a,b,c,d,e} = {0,X,X,X,1} or {X,0,X,X,1} or {X,X,0,0,1}.
  - Therefore, for node h, two test vectors are required to test for stuck-at fault.

Manufacturing Test Principles

– Example: Logic circuit in which a node is unobservable;



– Set h <= D, so that j <= D#.
– Set e <= 1, so that i <= D.
– But then k <= 0 **!** So that node **h** is unobservable.

Manufacturing Test Principles

- Fault Grading and Fault Simulation
  - Select node to be faulted,
  - Simulate with no faults inserted, then save results,
  - Fault the node (set to ‘0’ then to ‘1’),
  - Simulate again. If there exists difference between faulted response and saved response then fault is said to be detected and keep the test vector, else change the test vector and re-simulate.
- Delay Fault Testing
  - Functionally correct but fault may affect timing.
  - Hence, fault is sequential because detection of fault depends on the previous state of the gate and the simulation clock rate.

Manufacturing Test Principles

- Statistical Fault Analysis
  - Conventional fault analysis/simulation consumes large CPU resources and time.
  - Statistical fault analysis relies on estimating the probability that a fault will be detected.
- Fault Sampling
  - Nodes are randomly selected and faulted.

Design Strategies for Test

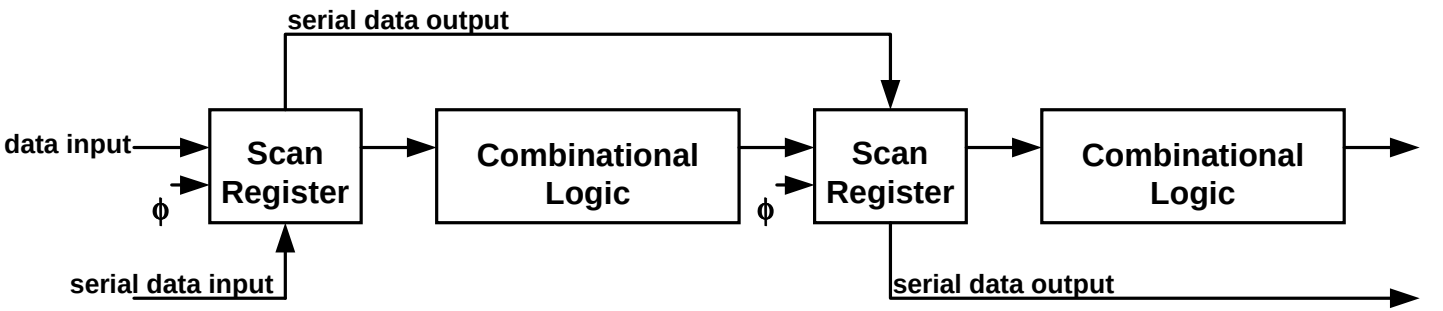
- Design For Testability (DFT)
  - Controllability and observability are the two key concepts for designing circuits that are testable.
    - Controllability is the ability to set every internal node to logic-1 and logic-0.
    - Observability is the ability to observe either directly or indirectly the state of any node in the circuit.
  - Three main approaches to Design For Testability
    - Ad-hoc (based on experience)
    - Scan based
    - Self-test and Built-in test

Design Strategies for Test

- Ad-Hoc Testing
  - Aim is to reduce complexity of testing. Common techniques are;
    - Partitioning large sequential circuits
    - Adding test points
    - Adding multiplexers to provide alternative signal paths
    - Providing for easy state reset
  - The techniques involve tricks that are developed over the years to avoid the overheads of systematic approach to testing.
  - Example: Consider a long counter (n-bit). Possible ad-hoc techniques are;
    - Add parallel-load feature to test carry propagation.
    - Reduce counter length to k bits, so that (n/k) 2^k output vectors to test instead of 2^n output vectors.
  - Example: Ina bus oriented system, the bus can be used for observing and controlling the internal nodes for testing.

Design Strategies for Test

- Scan-Based Test Techniques
  - Level Sensitive Scan Design (LSSD)



    - In this technique, every input combinational logic may be controlled and every output may be observed. Also, running a serial sequence of "110010" through scan registers can test these registers. The disadvantage is the complexity of the scan registers that impacts IC performance.
  - Serial Scan
  - Partial Serial Scan
  - Parallel Scan

Design Strategies for Test

- Self-Test (Built-in Test) Techniques

  Circuits perform operations on themselves in order to prove correct operation.

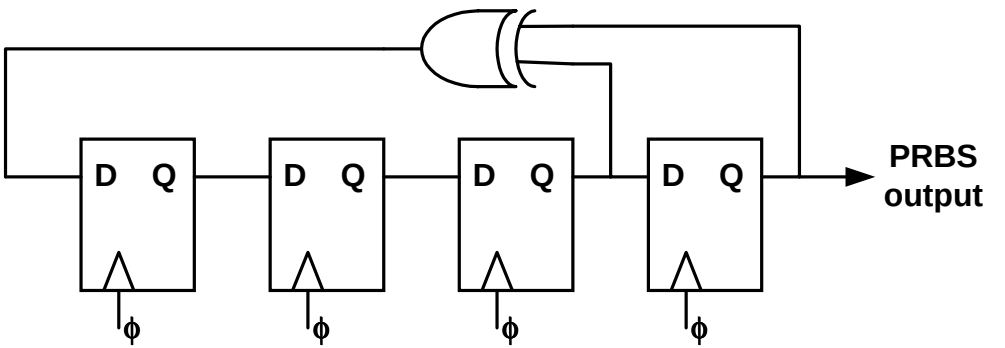
  – Signature Analysis and Built-in Logic Block Observation (BILBO)

  Uses a Pseudo-Random Sequence Generator (PRSG) / Pseudo-Random Binary Sequence (PRBS) to generate input signals for a section of combinational logic and then uses a Signature Analyzer to observe outputs.

Design Strategies for Test

- PRSG or PRBS generation: A linear feedback shift register (LFSR) is implemented by using an N-bit shift register with some of the outputs are ex-or'ed (addition) and fed back to the input. Choosing different otuputs for feedback results in different length and composition of binary sequence. The maximum-length sequence with N-bit shift register is $2^N - 1$ (all zero state is never entered).

  Example: A 4-bit LFSR that generated maximal-length sequence.

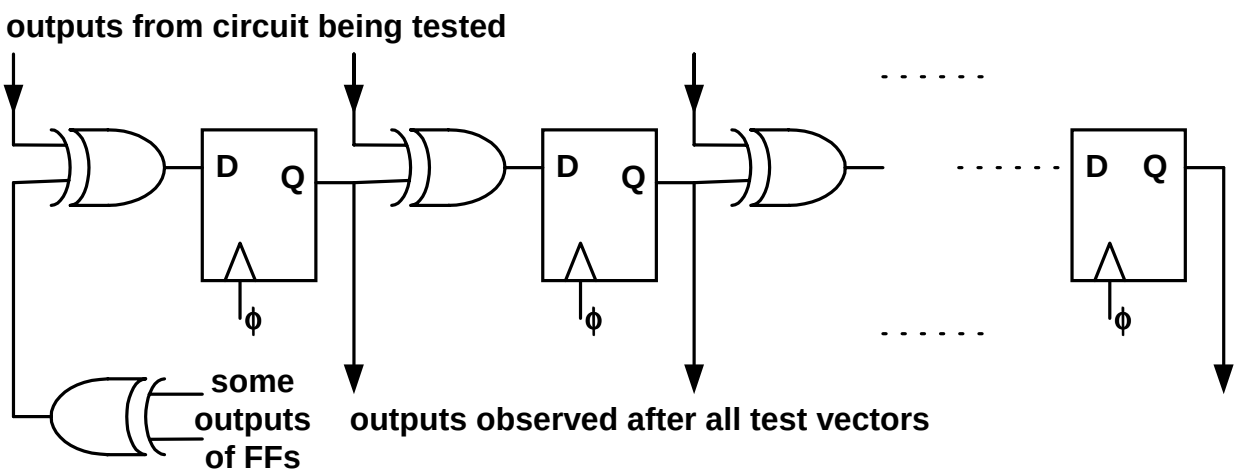


Output Sequence: ...111100010011010111100010011010111110001...

Design Strategies for Test

- Signature Analyzer: By modifying the LFSR circuit to ex-or an output from the circuit being tested, compaction of output data can be achieved. There is no need to store and compare every expected output from the circuit. Instead, one needs only to store and compare the expected final state (signature) of the LFSR for a fault-free circuit. For multi-output circuit, the outputs are ex-or'ed with several stages of the LFSR.

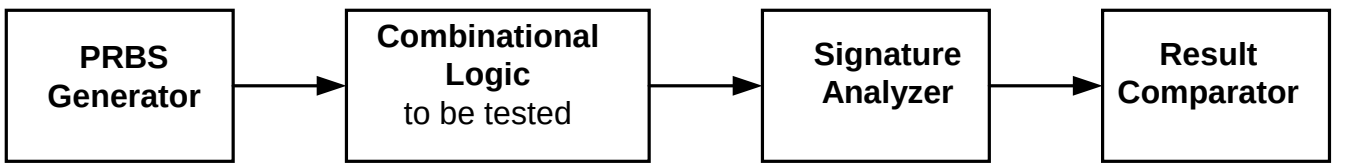
  Example:



Design Strategies for Test

- BILBO Structure: It is created when Signature Analysis is combined with the scan technique. So that, ex-or gates are included in to the LFSR and this register has the operation modes of
  - Scan
  - Reset
  - PRBS Generation / Signature Analysis
  - Parallel registers

  Example: BILBO used in a system.



– Memory Self-Test

Design Strategies for Test

- IDDQ Testing

  It is a method of testing for bridging faults. VDD current is monitored. Static CMOS logic gate does not draw DC current (except leakage). But, when bridging fault occurs, a measurable DC IDD flows. Current measuring is a slow process, hence test time increases.

Chip-Level Test Techniques

- Testing approach to the main types of circuit structure;
  - Regular logic arrays
    - Partial Serial Scan or Parallel Scan
  - Memories
    - Self-Testing
  - Random Logic
    - Full Serial Scan or Parallel Scan

System-Level Test Techniques

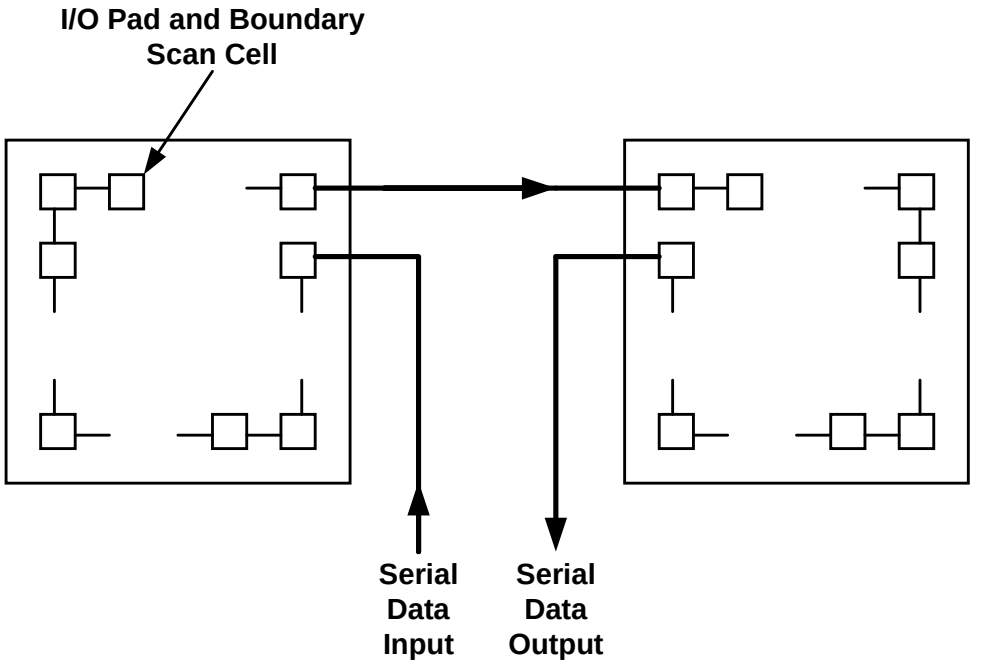
- At the board level
  - Bed-of-nails are used.
- At the chassis level
  - Software programs are used
- Complex boards
  - Scan-based test techniques
  - IEEE 1149 standard (Boundary Scan) is used to test multiple boards and IC's.

System-Level Test Techniques

- Boundary Scan (IEEE 1149 standard)
  - IC has a serial scan path through its I/O pins.
  - At the board level, IC's are connected in a variety of series and parallel combinations. Then the following types of tests can be done;
    - Connectivity tests among components
    - Sampling and setting chip I/O's
    - Distribution and collection of self-test or built-in test results.
  - IC has a Test Access Port (TAP). The port has the pins;
    - TCK : Test Clock Input
    - TMS : Test Mode Select
    - TDI : Test Data Input
    - TDO : Test Data Output
    - TRST : Test Reset Signal (Optional)

System-Level Test Techniques

- … Boundary Scan (IEEE 1149 standard)
  - The test circuitry is composed of
    - TAP pins
    - Test-data registers
    - Instruction register
    - TAP controller



  - Example: Boundary Scan Architecture

Layout Design for Improved Testability

- Can we construct the physical layouts to reduce likelihood of failures?
- Layout designer should be aware of nature of defects and the process.
- Open-Circuit fault immunity is increased by incorporating connection redundancy.